



**Уральский
федеральный
университет**

имени первого Президента
России Б.Н.Ельцина

**Институт радиоэлектроники
и информационных
технологий — РТФ**

И. А. СПИЦИНА

К. А. АКСЕНОВ

СИСТЕМНЫЙ АНАЛИЗ И МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Учебное пособие

Министерство науки и высшего образования
Российской Федерации

Уральский федеральный университет
имени первого Президента России Б. Н. Ельцина

**И. А. Спицина
К. А. Аксенов**

СИСТЕМНЫЙ АНАЛИЗ И МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Учебное пособие

Рекомендовано методическим советом
Уральского федерального университета
для студентов вуза, обучающихся
по направлению подготовки
09.04.01 — Информатика и вычислительная техника

Екатеринбург
Издательство Уральского университета
2021

УДК 303.732.4:681.518.001.57(075.8)

ББК 32.973.202я73

С40

Рецензенты:

кафедра «Шахматное искусство и компьютерная математика» Уральского государственного экономического университета (завкафедрой канд. экон. наук, доц. *Е. Н. Стариков*);
канд. техн. наук, доц. *С. С. Ланно* (генеральный директор АО «Ведение реестров компаний»).

Научный редактор — проф., д-р техн. наук *Л. Г. Доросинский*

Спицина, И. А.

С40 Системный анализ и моделирование информационных систем : учебное пособие / И. А. Спицина, К. А. Аксенов ; М-во науки и высшего образования РФ. — Екатеринбург : Изд-во Урал. ун-та, 2021. — 118 с.

ISBN 978-5-7996-3196-3

В издании рассмотрены этапы разработки информационных систем, начиная с планирования ИТ-проекта. Основное внимание уделено методам системного анализа, применяемым в процессе анализа предметной области, проектирования и разработки информационных систем. Описана технология проектирования программного обеспечения с использованием пакета BPsim. Пособие содержит примеры, иллюстрирующие материал. Предназначено для студентов дневной и заочной форм обучения.

Библиогр.: 17 назв. Табл. 6. Рис. 84.

УДК 303.732.4:681.518.001.57(075.8)

ББК 32.973.202я73

ISBN 978-5-7996-3196-3

© Уральский федеральный
университет, 2021

Оглавление

1. Особенности процесса разработки информационной системы	5
1.1. Основные понятия курса	7
1.2. Этапы проведения системного анализа	13
2. Планирование разработки информационной системы	20
2.1. Технические возможности	20
2.2. Экономические возможности	21
2.3. Организационные возможности	22
3. Анализ требований к информационной системе	26
3.1. Методы обследования предметной области	27
3.2. Методы определения узких мест в модели «как есть»	30
4. Моделирование процессов	35
4.1. Метод функционального моделирования SADT (IDEF0)	36
4.2. Методология IDEF3	38
4.3. Методология DFD	43
4.4. Нотация BPMN	46
5. Этап проектирования информационной системы	48
5.1. Дизайн архитектуры	49
5.2. Эксплуатационные требования	51
5.3. Требования к производительности	52
5.4. Требования к безопасности	54
5.5. Культурные и политические требования	55
6. Разработка пользовательского интерфейса	56
6.1. Базовые принципы разработки ПИ	57
6.2. Использование CASE-средства BPsim.SD при проектировании пользовательского интерфейса	60

7. Описание системы технико-экономического проектирования BPsim3	67
7.1. Назначение и логическая структура программы	67
7.2. Диаграммы и программы BPsim	70
7.2.1. Диаграммы DFD.....	70
7.2.2. Диаграммы прецедентов	74
7.2.3. Подсистема диаграммы классов	78
7.2.4. Диаграмма последовательности (поиска решения)	88
 8. Экспертное оценивание систем	100
8.1. Общая методика проведения экспертного оценивания...	100
8.2. Метод мозгового штурма	106
8.3. Метод Дельфи.....	108
8.4. Метод сценариев	110
8.5. Метод морфологического анализа	111
8.6. Экспертные игры	112
8.7. Применение методов экспертных оценок при разработке ИТ-проектов	113
 Библиографический список.....	115

1. Особенности процесса разработки информационной системы

Создание информационной системы (ИС) — долгосрочный сложный процесс, который сопровождается различными проблемами. Они связаны с возможными нарушениями сроков или бюджета проекта, с проблемами в команде разработчиков или качества готового программного продукта. Все это может привести к закрытию проекта или разработке информационной системы, которая не будет использоваться.

Увеличить вероятность успешного завершения проекта помогут фундаментальные концепции, технологии и методы, представленные в этом учебном пособии.

Ключевую роль в процессе разработки ИС играет системный аналитик. Для его работы необходимы следующие навыки [1]:

- технические — понимание существующей технической среды предприятия, основ новых технологий и способы их применения;
- деловые — понимание, как информационные технологии (ИТ) могут быть применены к текущей бизнес-ситуации, гарантирование того, что их использование поможет решить проблемы бизнеса;
- аналитические — регулярно используются при решении проблем, возникающих в ИТ-проекте, а также связанных с организационным уровнем;
- межличностные — постоянное общение со многими людьми: пользователями, менеджерами, программистами, которые имеют разный опыт в разных областях;
- управленческие — управление людьми, с которыми он работает, управление рисками и принятие управленческих решений в условиях неопределенности данных;

- этические — этические вопросы при общении с коллегами и вопросы, связанные с обладанием конфиденциальной информации.

Жизненный цикл ИС состоит из четырех этапов: планирования, анализа, проектирования и реализации. Рассмотрим работу аналитика на каждом этапе.

Этап планирования является самым фундаментальным этапом, на котором определяются причины и способы создания ИС. Аналитик должен оценить следующие аспекты:

- технические возможности — можно ли разработать ИС с необходимыми функциями;
- экономические возможности — приведет ли разработка ИС к увеличению дохода предприятия;
- организационные возможности — достаточно ли у предприятия ресурсов для использования новой системы.

По окончании данного этапа необходимо подготовить требования к системе и технико-экономическое обоснование, план проекта.

На этапе анализа необходимо определить потенциальных пользователей ИС, ее основные функции, а также где и когда она будет использоваться. На данном этапе необходимо провести анализ имеющихся ИС, определить показатели, которые следует улучшить, и разработать концепцию новой системы. На нем выполняются следующие действия:

- стратегический анализ. Изучение системы «как есть», ее проблем, определение путей по проектированию системы «как будет»;
- сбор требований (интервью, опросы, групповые семинары). Анализ полученной информации. Разработка концепции системы. Разработка бизнес-моделей системы, которые показывают, как будет использоваться ИС в бизнес-процессах предприятия;
- разработка требований к системе.

На этапе проектирования выполняются следующие действия:

- определяется стратегия проектирования — самостоятельная разработка, разработка сторонней компанией или покупка существующего программного обеспечения;
- проектирование архитектуры системы и пользовательского интерфейса (основных форм и отчетов);
- проектирование структуры базы данных и служебных файлов;
- проектирование ИС — определение модулей ИС и их функций.

В результате работы готовят спецификацию системы.

На этапе реализации аналитик разрабатывает план поддержки ИС, который включает в себя формальные и неформальные обзоры достоинств и недостатков новой системы.

Краткий обзор этапов разработки показывает, что аналитик принимает активное участие на каждом из них.

1.1. Основные понятия курса

Термины «анализ» и «синтез» широко используются в разных науках: математике, химии, философии. В общем случае они определяются следующим образом [2]:

- анализ — прием мышления, связанный с разложением изучаемого объекта на составные части, стороны, тенденции развития и способы функционирования для их относительно самостоятельного изучения;
- синтез — противоположная анализу операция, заключающаяся в объединении ранее выделенных частей в целое для получения знаний о целом путем выявления тех существенных связей и отношений, которые объединяют ранее выделенные в анализе части в одно целое.

При исследовании систем используют, помимо всего, агрегирование и декомпозицию.

Агрегирование — процедура объединения частей в целое.

Декомпозиция — процедура разложения целого на части.

Применение анализа при исследовании системы подразумевает процесс декомпозиции с последующим определением статических и динамических характеристик элементов, рассматриваемых во взаимосвязи с другими элементами систем и окружающей средой.

Цели анализа информационной системы следующие:

- детальное изучение системы для более эффективного использования и принятия решения по ее дальнейшему совершенствованию или замене;
- исследование альтернативных вариантов вновь создаваемой информационной системы в целях выбора наилучшего варианта.

К задачам анализа системы относятся:

- определение объекта анализа;
- структурирование системы;
- определение функциональных особенностей системы;
- исследование информационных характеристик системы;
- определение количественных и качественных показателей системы;
- оценка эффективности системы;
- обобщение и оформление результатов анализа.

Таким образом, при изучении системы можно выбрать одно из двух направлений анализа: определение состояния системы, чтобы выявить части, требующие улучшения, и стимулирование изменений или исследование альтернативных вариантов вновь создаваемой системы для выбора лучшего варианта.

Синтез информационной системы — это процесс создания новой системы путем определения ее рациональных или оптимальных свойств и соответствующих показателей [17].

Возможен синтез информационной системы с целью:

- создания новой ИС, которая будет использовать современные ИТ-технологии;
- модернизации существующей ИС для устранения проблем, выявленных при работе, или добавления функций, связанных с появлением новых бизнес-процессов и требований.

Можно выделить следующие основные задачи синтеза ИС:

- определение структуры и параметров ИС на основе требований и ограничений, сформулированных для нее и связанных с условиями функционирования ИС;
- определение способов разработки ИС.

Различают структурный и параметрический синтез и их комбинацию. Синтез представляет собой процесс разработки ИС, удовлетворяющую заданным требованиям.

В начале раздела описаны проблемы, связанные с разработкой ИС. Кроме того, тенденции развития современных информационных технологий приводят к росту сложности проектов по созданию ИС. Можно выделить следующие особенности крупных современных проектов [3]:

- сложность описания — необходимо реализовать большое количество функций, процессов, которые используют разнообразные данные и имеют сложные взаимосвязи друг с другом, для этого

требуется скрупулезное моделирование и анализ данных и процессов;

- сложность структуры — наличие большого числа взаимодействующих подсистем, имеющих свои локальные задачи и цели функционирования;
- особенности процессов, которые ограничивают возможность использования различных типовых проектных решений и прикладных систем;
- необходимость интеграции существующих и вновь разрабатываемых ИС;
- неоднородная среда функционирования ИС, использование нескольких аппаратных платформ;
- неоднородность отдельных групп разработчиков по уровню квалификации и использованию тех или иных средств разработки;
- существенная временная протяженность проекта, обусловленная, с одной стороны, ограниченными возможностями коллектива разработчиков, и, с другой стороны, масштабами организации-заказчика и различной степенью готовности отдельных ее подразделений к внедрению ИС.

Уменьшить влияние этих особенностей на результаты проектирования можно за счет использования системного анализа и системного подхода.

Системный анализ представляет собой достаточно новую научную дисциплину, которая может применяться в любой предметной области. Далее он будет рассматриваться в контексте вопросов разработки информационных систем. При изложении основ анализа и синтеза ИС используется понятийно-содержательный подход, заключающийся во внимании к основным понятиям, идеям, концепциям, которые основаны на методологических принципах. Разъяснение материала на конкретных примерах позволяет лучше понять суть принципов.

Рассмотрим основные понятия системного анализа [4].

Системный анализ — научная дисциплина, занимающаяся проблемой принятия решения в условиях анализа большого количества информации. Из определения следует, что целью системного анализа проблемы является повышение степени обоснованности принимаемого решения, расширение множества вариантов, среди которых производится выбор с одновременным указанием способов отбрасывания, заведомо уступающих другим вариантам.

Системный анализ в широком смысле — это методология (совокупность методических приемов) постановки и решения задач построения и исследования систем, тесно связанная с математическим моделированием. В более узком смысле системный анализ — методология формализации сложных (трудноформализуемых, плохо структурированных) задач.

Системный подход — исследования, проводимые комплексно, с разных сторон изучения предмета или явления. Системный подход предполагает, что все частные задачи, решаемые на уровне подсистем, должны быть увязаны между собой и решаться с позиции целого (принцип системности).

Принципами системного подхода называются положения общего характера, являющиеся обобщением работы человека со сложными системами (ядро методологии). Известно около двадцати таких принципов, ниже приведем основные [4]:

- принцип конечной цели — абсолютный приоритет конечной цели над частными. При анализе системы это поможет выявить ее основные свойства, показатели качества и критерии оценки. При синтезе системы необходимо каждый раз оценивать достижимость конечной цели;
- принцип единства — совместное рассмотрение системы как целого и как совокупности элементов. При разделении системы на части всегда необходимо сохранять представление о ней, как о едином целом;
- принцип связности — рассмотрение любой части системы совместно с ее окружением;
- принцип модульного построения — необходимо выделить в системе отдельные модули и рассматривать ее как совокупность модулей. Таким образом можно изучать входные и выходные воздействия системы;
- принцип иерархии — полезно выделить иерархию элементов;
- принцип функциональности — совместное рассмотрение структуры и функций с приоритетом функций над структурой;
- принцип развития — учет изменяемости системы, ее способности к развитию, расширению, замене частей и накоплению информации;
- принцип децентрализации — сочетание в принимаемых решениях централизации и децентрализации;

- принцип неопределенности — учет случайностей и неопределенностей в системе.

Системный подход способствует выработке правильного метода мышления о процессе разработки ИС, поскольку ИС применяется на предприятии, которое является частью большей системы, и его цели функции постоянно претерпевают изменения. Если в процессе разработки у команды нет достаточной информации о существе проблемной ситуации, то для того чтобы организовать процесс принятия решений, применяется системный анализ.

Любая задача системного анализа начинается с построения модели исследуемой системы [4]. Для этого необходимо сначала изучить структуру системы, выполнить анализ ее компонентов, выявить взаимосвязи между отдельными элементами. Чтобы обоснованно проводить анализ структуры системы, необходимо рассмотреть ряд понятий и определений, характеризующих строение и функционирование системы.

Под системой понимают совокупность элементов, находящихся в отношениях и связях друг с другом и со средой, образующих определенную целостность, единство.

Элементом называют простейшую неделимую часть системы при данном способе расчленения.

Подсистема — совокупность взаимосвязанных элементов, обладающих свойствами (в частности, свойством целостности), способная выполнять относительно независимые функции, иметь подцели, направленные на достижение общей цели системы. Отличие подсистемы от простой группы элементов состоит в том, что для подсистемы формулируются подцели ее функционирования.

Если части систем не обладают свойством целостности и способностью выполнять независимые функции, а представляют собой совокупность однородных элементов, то такие части называют компонентами.

Структура отражает определенные взаимосвязи, взаиморасположение составных частей системы, ее устройство, строение.

Связь — ограничение степени свободы элементов. Связь характеризуется направлением, силой и характером. Связи бывают: направленные и ненаправленные; сильные и слабые; связи подчинения, равноправные, генетические, связи управления. Различают также связи по направленности процессов — прямые и обратные.

Обратные связи могут быть положительными, т. е. сохранять происходящие в системе изменения того или иного параметра, и отри-

цательными, т. е. противодействующими тенденциям изменения выходного параметра. Обратная связь является основой приспособления систем к изменениям существования, основой саморегуляции и развития системы.

Внутренняя среда — совокупность объектов, которые находятся в пределах границ системы, влияют на ее поведение, но не принадлежат ей.

Внутренняя среда каждой организации формируется под воздействием переменных, оказывающих непосредственное влияние на процесс преобразований (производства продукции и услуг). Она состоит из следующих внутренних переменных:

- цели;
- структуры;
- задач;
- технологии;
- людей.

Изменение в одной из них в определенной степени влияет на все остальные. Совершенствование одной переменной, например технологий, необязательно может привести и к повышению производительности труда, если эти изменения сказываются отрицательно на другой переменной — людях.

Внешняя среда — окружающая среда системы, или совокупность объектов, которые располагаются за границами системы, воздействуют на нее, но не принадлежат системе.

Внешняя среда организации состоит из двух компонентов:

- среды прямого воздействия, которая включает —
 - ✓ поставщиков;
 - ✓ государственные органы;
 - ✓ профсоюзы;
 - ✓ конкурентов;
 - ✓ потребителей;
- среды косвенного воздействия, которая складывается —
 - ✓ из международного окружения;
 - ✓ научно-технического прогресса;
 - ✓ политических факторов;
 - ✓ социокультурных факторов;
 - ✓ состояния экономики.

Вход системы — связь системы с окружающей средой, направленная от среды к системе.

Выход системы — связь системы с окружающей средой, направленная от системы к среде.

Ограничение системы — предел, внутри которого объекты, свойства и их связи можно адекватно объяснить и обеспечить управление ими.

Надсистема (суперсистема) — часть внешней среды, для которой исследуемая система является элементом.

Свойства системы:

- каждая часть системы обладает свойствами, которые она теряет в случае отделения от системы;
- каждая система обладает определенными (существенными) свойствами, которыми не обладает ни одна из ее частей.

Процесс разработки новой информационной системы или модернизация существующей должен начинаться с анализа внутренней и внешней среды организации, с изучения их влияния на систему.

1.2. Этапы проведения системного анализа

Существуют различные варианты проведения системного анализа. Одна из возможных последовательностей этапов приводится ниже.

- Анализ состояния внешней и внутренней среды и прогноз их изменения, выявление аспектов, которые оказывают влияние на разрабатываемую систему.
- Формирование целей, дерева целей.

На данном этапе следует определить глобальную цель, для чего разрабатывается новая или модернизируется существующая ИС. Обычно целью проектирования ИС является получение прямого или косвенного экономического эффекта от ее внедрения. Для декомпозиции цели используется метод дерева целей, описанный ниже.

- Формирование проблемных ситуаций (при которых цели не достигнуты). Нужно подобрать конкретные примеры — что может повлиять на работу конкретной системы.
- Формирование критериев, предпочтений, ограничений.
- Формирование альтернативных ситуаций. Описание альтернативных ситуаций должно содержать краткое описание систем-аналогов, если они рассматриваются в виде альтернативы.

Должен быть описан способ сравнения альтернатив, например бюджет или сроки выполнения проекта.

- Выбор наилучшей альтернативы.

Метод дерева целей — метод, который основан на принципе разделения общей цели создания ИС на подцели, которые в свою очередь делятся на цели нижележащих уровней. Использование данного метода перед разработкой ИС имеет большое значение, поскольку четко формулируются цели создания ИС и способы их достижения.

Цели, которые предполагает достичь организация за счет нового программного продукта, можно классифицировать по следующим признакам:

- источникам возникновения —
 - ✓ навязанные требованиями внешней среды предприятия, например требованиями законодательства;
 - ✓ возникающие в результате необходимости удовлетворения потребностей внутренней среды предприятия, например повышения производительности сотрудников;
- с точки зрения комплексности —
 - ✓ простые;
 - ✓ сложные, которые декомпозируются на подцели;
- степени важности —
 - ✓ стратегические, которые ставятся для решения перспективных масштабных проблем;
 - ✓ тактические, которые разрабатываются для достижения стратегических целей;
- содержанию —
 - ✓ технологические, которые ставятся для внедрения новых ИТ-технологий на предприятии;
 - ✓ экономические, которые ставятся для повышения финансовых результатов;
 - ✓ производственные, которые ставятся для внедрения новых услуг или повышения их качества;
- с точки зрения приоритетности —
 - ✓ необходимые, которые обеспечивают функционирование организации;
 - ✓ желательные, достижение которых благоприятно скажется на организации;

- ✓ возможные, которые никак не скажутся на существовании и развитии организации в настоящее время;
- направленности —
 - ✓ на конечный результат, например предоставление новой услуги;
 - ✓ осуществление той или иной деятельности, например увеличение количества обслуживаемых клиентов в единицу времени;
- форме выражения —
 - ✓ выраженные в количественных показателях;
 - ✓ описанные качественными характеристиками;
- с точки зрения особенностей взаимодействия —
 - ✓ индифферентные — цели, безразличные по отношению друг к другу;
 - ✓ конкурирующие;
 - ✓ комплементарные — цели, которые дополняют друг друга;
 - ✓ антагонистические — цели, которые исключают друг друга;
 - ✓ идентичные — совпадающие.

Процесс построения дерева целей состоит из следующих этапов:

- анализа стратегии предприятия и формулировки глобальной цели разработки или модернизации ИС;
- анализа способов достижения цели верхнего уровня и формулировки подцели;
- проверки независимости подцелей, их осуществимости, при необходимости уточнения их формулировок;
- проверки элементарности подцелей. Если цели можно еще декомпозировать, то следует повторить п. 2 (анализ способов достижения цели) — 4 (данный пункт);
- построения дерева целей.

При построении дерева целей необходимо руководствоваться следующими правилами:

- каждая сформулированная цель должна иметь средства и ресурсы для ее обеспечения;
- при декомпозиции целей должно соблюдаться условие полноты редукции, т. е. количество подцелей каждой цели должно быть достаточным для ее достижения;
- декомпозиция каждой цели на подцели осуществляется по одному выбранному классификационному признаку;
- развитие отдельных ветвей дерева может заканчиваться на разных уровнях системы;

- вершины вышележащего уровня системы представляют собой цели для вершин нижележащих уровней;
- развитие дерева целей продолжается до тех пор, пока лицо, решающее проблему, не будет иметь в распоряжении все средства для достижения вышестоящей цели.

Приведем пример системного анализа перед разработкой системы автоматизации проведения онлайн-конференций.

- Анализ состояния внешней и внутренней среды.

Объектом автоматизации является учебная организация, занимающаяся проведением образовательных онлайн-конференций.

Основным автоматизируемым процессом является организация и непосредственное проведение конференции.

Внутренней средой по отношению к рассматриваемому объекту автоматизации являются сотрудники организации, занимающиеся подготовкой и проведением конференции, и оборудование, необходимое для этого, (рис. 1.1).

К внешней среде относятся посетители конференции, спикеры конференции, которые не являются сотрудниками организации, спонсоры, конкуренты — предприятия, предоставляющие аналогичные услуги (рис. 1.1).

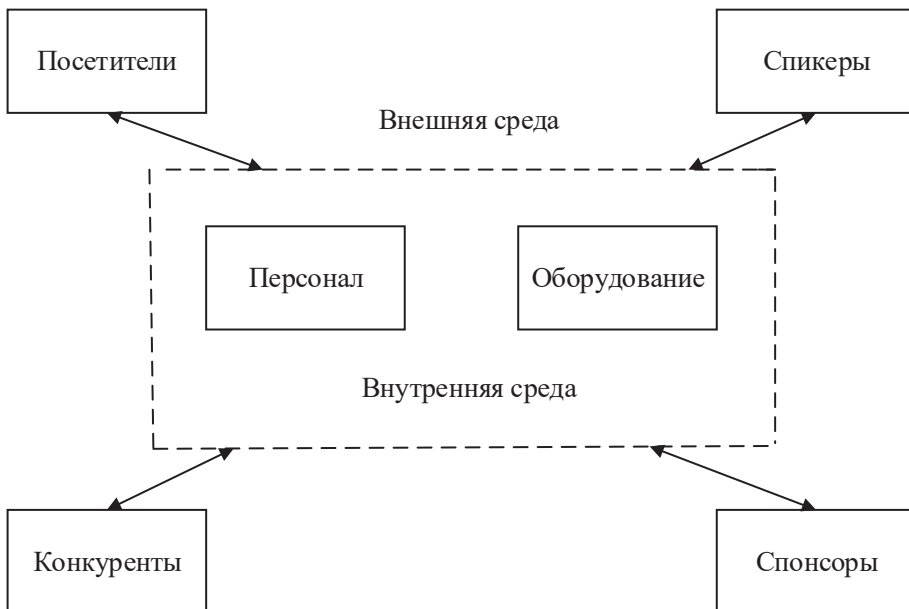


Рис. 1.1. Внешняя и внутренняя среда системы

- Прогноз изменений внешней и внутренней среды.

Прогноз изменений внешней и внутренней среды представлен в табл. 1.1.

Таблица 1.1

Прогноз изменений внешней и внутренней среды

Среда	Элемент	Влияние	Меры	Критичность
Внутренняя	Персонал	Потребности и квалификация персонала влияют на функции системы и пользовательский интерфейс	Проведение анализа потребностей персонала. Согласование прототипа системы	Средняя
	Оборудование	Разрабатываемая система должна соответствовать требованиям оборудования предприятия. Возможны его поломки	Разворачивание распределенной гибкой отказоустойчивой системы. Мониторинг и увеличение ресурсов по мере необходимости	Высокая
Внешняя	Посетители и спикеры	Их потребности могут привести к доработкам системы	Проведение анализа потребностей до начала разработки	Средняя
	Конкуренты	Более удобные платформы у конкурентов приведут к уменьшению количества клиентов и спикеров	Проведение анализа конкурентов перед началом разработки, мониторинг в процессе разработки и эксплуатации системы	Высокая
	Спонсоры	Контрагенты могут менять условия, повышать ценники на обслуживание и услуги	Мониторинг состояния рынка. Вложение средств в амортизацию	Высокая

Примечание. Вероятность изменения сред оценивается как средняя.

- Дерево целей.

Дерево целей этого проекта представлено на рис. 1.2.

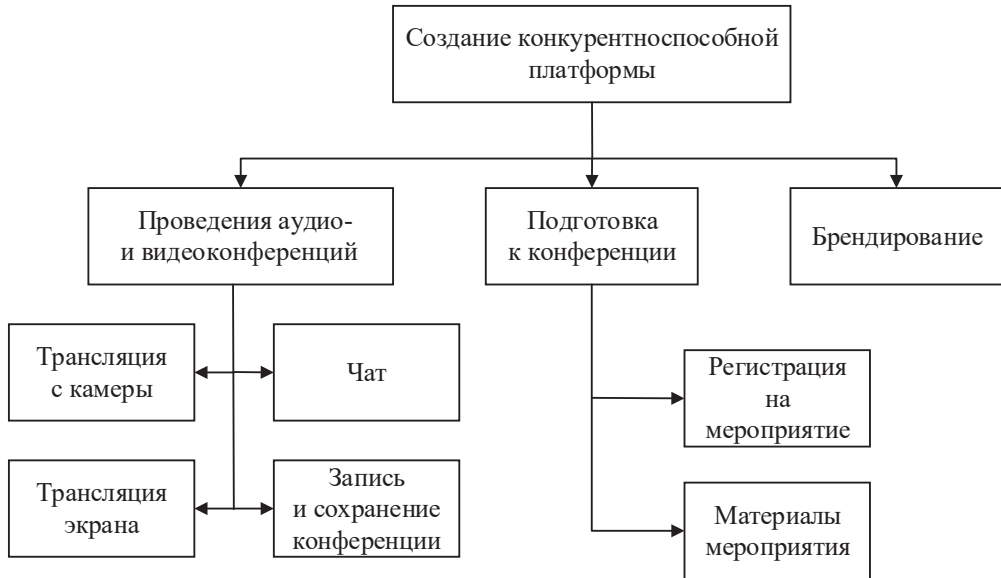


Рис. 1.2. Дерево целей

- Проблемные ситуации.

Во время эксплуатации системы могут возникать проблемные ситуации, из-за которых цели могут быть не достигнуты. Можно выделить следующие ситуации:

- ✓ технические сбои;
- ✓ неадекватное поведение участников конференции;
- ✓ увеличение стоимости услуг контрагентов;
- ✓ появление новых требований у различных пользователей информационной системы.

Для снижения рисков или исключения данных проблем описываемые ситуации следует учесть как требования, предъявляемые к информационной системе.

- Критерии, предпочтения, ограничения.

Исходя из определенных целей и проблем можно выделить следующие критерии и ограничения для выбора альтернативных решений:

- ✓ количество участников;
- ✓ возможность организовать подготовку к конференции;

- ✓ брендинг;
- ✓ возможность бесплатного использования.
- Выбор наилучшей альтернативы.

Сравнительный анализ возможных вариантов — разработка собственной системы или использование готовых продуктов — представлен в табл. 1.2.

Таблица 1.2

Сравнительный анализ альтернативных вариантов

Критерий	Альтернатива		
	Разработка собственной системы	Zoom	GetCourse
Максимальное количество слушателей	Неограниченное	100	До 5000
Возможность организовать подготовку к конференции	Да	Нет	Нет
Брендинг	Да	Нет	Нет
Возможность бесплатного использования	Да	Да	Нет

Результаты сравнения альтернативных вариантов получения готовой системы по критериям и ограничениям показывают, что наилучшей альтернативой является собственная разработка.

2. Планирование разработки информационной системы

В гл. 1 было показано, что на этапе планирования разработки информационной системы необходимо провести технико-экономический анализ. Он позволяет оценить, следует ли выполнять этот проект и риски, связанные с проектом. Техничко-экономический анализ включает в себя три составляющих: техническую, экономическую и организационную [1].

2.1. Технические возможности

Анализ технических возможностей позволяет оценить риски ИТ-проекта, связанные с технической составляющей, т.е. позволяет ответить на следующие вопросы:

- возможно ли самостоятельно разработать информационную систему;
- возможно ли, целесообразно ли приобретение готовой системы;
- будет ли новая система успешно внедрена и использована на предприятии.

Анализ технических возможностей включает в себя несколько этапов.

- Понимание бизнес-процессов предприятия.

Если аналитик не знаком с предметной областью, то существует большая вероятность неправильно интерпретировать информацию о бизнес-процессах, которая получена от пользователей, или невозможность предложить улучшения. Ситуация еще более ухудшается, если пользователи плохо представляют существующие процессы и не могут их детально описать. В таком случае предпочтительнее рассматривать проект о расширении существующей информационной системы, потому что она понятнее пользователям.

- Знание технологий.

Если при разработке информационной системы предполагается использовать технологии, которые ранее не использовались командой разработчиков, то существует вероятность увеличения сроков проекта, связанного с изучением новых технологий. Также возможны проблемы при применении технологий, которые недавно появились, поскольку они мало изучены и описаны.

- Размер проекта.

Необходимо оценить количество разработчиков и функций системы. Большим проектом сложнее управлять, повышается риск, если не учесть какие-то требования.

- Совместимость.

Если новая система будет интегрироваться с уже существующими на предприятии, то это увеличивает сложность проекта, а следовательно, и повышает риски. Необходимо оценить совместимость новой системы с уже имеющимися на предприятии.

Оценка технических возможностей может быть неоднозначной, поскольку во многих случаях одно и то же может интерпретироваться по-разному. Для повышения ее адекватности, можно сравнить проекты с уже сделанными или проконсультироваться с ИТ-экспертами.

2.2. Экономические возможности

Анализ экономических возможностей включает в себя определение:

- стоимости разработки;
- ежегодных эксплуатационных затрат;
- общей годовой прибыли;
- нематериальных выгоды и затрат.

Этапы проведения анализа экономических возможностей следующие.

- Определение расходов и доходов.

Необходимо составить список всех расходов и доходов. Записать их, разделив на группы: производственные (зарплата разработчиков, плата за консультацию) и операционные расходы (покупка необходимого программного обеспечения и оборудования), материальные (со-

крашение персонала, увеличение доходов от основных услуг предприятия) и нематериальные доходы (увеличение доли рынка, повышение качества продукции или услуг).

- Оценка размеров расходов и доходов.

При необходимости следует проконсультироваться со специалистами и оценить все расходы и доходы, даже нематериальные.

- Определение денежного потока.

Следует спрогнозировать на 3–5 лет, что больше: расходы или доходы.

- Оценить экономическую эффективность проекта.

Для оценки экономической эффективности ИТ-проекта чаще всего применяются три основных финансовых метода [5]:

- ✓ NPV (Net present value) — чистый приведенный доход, или чистая приведенная стоимость;
- ✓ IRR (Internal rate of return) — внутренняя норма доходности, или внутренняя норма рентабельности;
- ✓ Payback (Break-Even Point) — срок окупаемости инвестиций.

2.3. Организационные возможности

Анализ организационных возможностей показывает, насколько хорошо система в конечном счете будет принята пользователями и встроена в существующие бизнес-процессы предприятия. Существует много организационных факторов, влияющих на оценку системы.

Один из путей оценки организационных возможностей проекта — это анализ соответствия целей проекта целям бизнеса. Проекты, которые инициируются только ИТ-отделом, нередко так и не начинают внедряться на предприятии.

Для достижения высокой степени соответствия между бизнесом и ИТ, Дж. Хендерсоном и Н. Венкатраманом была предложена модель стратегического выравнивания [8]. Стратегическое выравнивание — приведение в соответствие бизнес- и информационной стратегий [6, 7]. Она позволяет анализировать связи между областями бизнеса и ИТ как на стратегическом уровне, так и на операционном.

Основные элементы стратегической модели представлены на рис. 2.1.

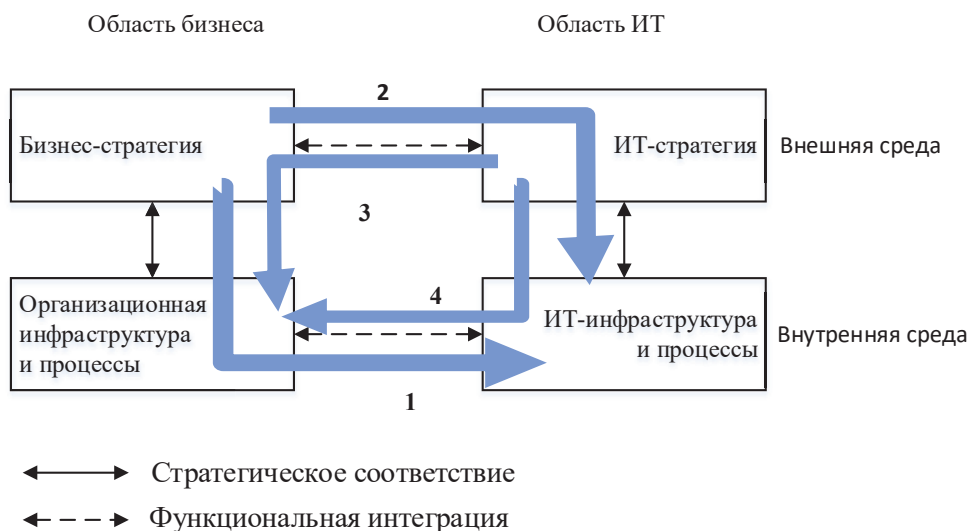


Рис. 2.1. Модель стратегического выравнивания

В стратегической модели выделяют две области — бизнес и ИТ, которые рассматривают аспекты, относящиеся к внутренней или внешней для организации среде. Бизнес-стратегия — это представление предприятия во внешней конкурентной среде: его позиция на рынке, конкурентные преимущества, отношения с партнерами. Организационная инфраструктура и процессы — это описание организационной структуры, важнейших бизнес-процессов, а также решения, связанные с приобретением и развитием навыков, необходимых для получения требуемых организационных компетенций [7]. ИТ-стратегия — это представление организации на внешнем рынке ИТ: технологии, которые поддерживают существующие или способствуют появлению новых бизнес-инициатив, конкурентные преимущества с точки зрения ИТ (надежность и гибкость системы, информационная безопасность), приобретение новых необходимых ИТ-компетенций. ИТ-инфраструктура и процессы — это описание решений, которые определяют системную архитектуру и ключевые ИТ-процессы, а также решения, связанные с приобретением и развитием знаний, необходимых для эффективного управления технической инфраструктурой в рамках организации [7].

Области модели связаны двумя типами связей: стратегическим соответствием (вертикальная стрелка на рис. 2.1) и функциональной ин-

теграцией (горизонтальная стрелка на рис. 2.1). Стратегическое соответствие определяет необходимость учитывать состояние внутренней и внешней среды при разработке новой стратегии. Функциональная интеграция между бизнесом и ИТ показывает влияние решений, принимаемых в одной области, на другую. На стратегическом уровне она отражает способность ИТ-стратегии изменять и поддерживать бизнес-стратегию. На операционном уровне она показывает важность существования внутренней согласованности между организационными требованиями и возможностями ИТ.

Авторы модели предложили четыре общих направления или перспективы стратегического выравнивания бизнеса и ИТ [8].

- Исполнение стратегии (стрелка 1 на рис. 2.1).

Данное направление позволяет реализовать идеи руководства предприятия. ИТ-инфраструктура просто внедряет эти предложения. В таком случае она не рассматривается в качестве конкурентного преимущества предприятия, решения принимаются исходя из минимизации затрат на ИТ.

- Технологический потенциал (стрелка 2 на рис. 2.1).

Данное направление рассматривает ИТ-службу в качестве бизнес-партнера, который способен реализовать предложения руководства предприятия, сформулировав логику ИТ-стратегии, а затем модернизировать ИТ-инфраструктуру и процессы.

- Конкурентный потенциал (стрелка 3 на рис. 2.1).

Это направление касается использования новых информационных технологий для модернизации бизнес-стратегии. ИТ-служба определяет, какие новые тенденции на рынке позволяют предприятию получить конкурентные преимущества, основанные на предоставлении новых товаров или услуг.

- Уровень услуг (стрелка 4 на рис. 2.1).

Такое направление не затрагивает бизнес-стратегию организации. ИТ-служба выполняет роль поставщика услуг для бизнеса, поэтому наиболее важной становится ее операционная эффективность.

Другой путь оценки организационных возможностей проекта — анализ заинтересованных лиц, под которыми понимаются люди, группы или организации, которые влияют на новую систему или на которые она влияет.

Перечислим основных заинтересованных лиц:

- ✓ спонсор проекта;

- ✓ менеджмент организации;
- ✓ пользователи системы.

Спонсор проекта предоставляет финансирование, политическую поддержку, поэтому лучше иметь несколько спонсоров. Спонсор проекта формирует базовые системные требования. Для усиления его вовлеченности следует показать ему презентацию о целях и задачах проекта и предполагаемых преимуществах новой системы. Позднее продемонстрировать ему прототип системы для демонстрации ее потенциальных возможностей.

Поддержка проекта руководством организации и руководителями среднего звена показывает остальным сотрудникам, что новая система внесет ценный вклад в развитие предприятия и что они найдут необходимые ресурсы для ее внедрения. Для их привлечения также нужна демонстрация презентации о целях и задачах проекта и предполагаемых преимуществах новой системы; необходимо подготовить небольшие информационные брошюры с описанием преимущества новой системы, по возможности проводить встречи спонсоров и руководства организации для обмена мнениями.

Пользователи системы также должны принимать активное участие в проекте. В конечном итоге именно они определяют, является ли проект успешным, поскольку именно они используют или не используют новую систему. На каждом этапе необходимо получать от пользователей обратную связь в отношении проекта.

Финальное технико-экономическое обоснование поможет организации принять правильное решение относительно новой ИС, потому что команда разработчиков рассмотрела все факторы, влияющие на проект.

3. Анализ требований к информационной системе

На этапе анализа требований к ИС аналитик работает с пользователями, чтобы понять их требования к новой системе. Не стоит надеяться на то, что пользователи сформируют требования к системе, которые затем сразу можно использовать. Аналитик должен критически подойти к полученной информации и переработать ее в требования к системе [1]. Процесс анализа включает в себя три этапа:

- построение модели системы «как есть»;
- определение узких мест и, возможно, реинжиниринг бизнес-процессов;
- построение модели системы «как будет».

Следует отметить, что в случае применения RAD-технологии или Agile не придается большое значение первому этапу. Однако опыт разработки показывает, что изучение текущего состояния бизнес-процессов важно для конечного результата проекта.

Определение требований к ИС выполняется для преобразования высокоуровневых бизнес-требований к системе в более детальный и точный перечень требований. Требование — это заявление о том, что система должна делать и какие характеристики она должна иметь. Требования бывают следующих видов: требования бизнеса, требования пользователя, функциональные требования, нефункциональные требования (производительность, интерфейс пользователя) и системные требования.

Приведем пример преобразования одних видов требований в другие.

Руководитель говорит, что введение новой ИС должно позволить предприятию расширить свою долю на рынки; уменьшить стоимость обслуживания клиентов. Пользователи системы хотят иметь возможность просматривать расписание (когда, куда, какой клиент записан). Аналитик формирует следующие функциональные требования: ввод

клиентами своих заявок на сайте, просмотр введенных заявок, назначение выполняющих заявки, получение отчета. Таким образом аналитик определяет необходимую информацию и функции системы.

3.1. Методы обследования предметной области

Можно выделить следующие методы обследования предметной области:

- интервью;
- совместную разработку приложений;
- анкетирование;
- анализ документов;
- наблюдение.

Ниже кратко опишем эти методы.

- **Интервью.**

Наиболее часто используемый способ — интервью. Для его проведения необходимо:

- ✓ определиться с кругом опрашиваемых лиц;
- ✓ придумать вопросы для интервью;
- ✓ провести его;
- ✓ обработать полученную информацию.

В круг опрашиваемых лиц нужно включить людей из разных уровней организации (от высшего руководства до конечных исполнителей) и с разными точками зрения. Такой подход позволит получить общую картину бизнес-процессов предприятия.

При проведении интервью используют три типа вопросов [1]:

- ✓ закрытый — это вопрос, на который мы ожидаем получить только два варианта ответов: либо «да», либо «нет»;
- ✓ открытый — это вопрос, на который мы ожидаем получить максимально развернутый и полный ответ по заданной в вопросе теме;
- ✓ уточняющий — этот вопрос задается для уточнения обсуждаемой информации.

Цели закрытого вопроса:

- ✓ получить согласие;
- ✓ получить подтверждение договоренностей или предположений;

- ✓ уточнить информацию;
- ✓ зафиксировать что-либо;
- ✓ завершить разговор.

Цели открытого вопроса:

- ✓ разговаривать пользователя;
- ✓ получить общую информацию;
- ✓ понять, что стоит за утверждениями пользователя;
- ✓ получить паузу для обдумывания.

Не существует каких-то единых рецептов для подготовки к интервью. Можно предложить два способа организации вопросов для интервью: от общего к частному и от частного к общему. На начальном этапе сбора информации, когда очень многое неясно, можно заранее подготовить несколько открытых вопросов, а уточняющие вопросы придумывать по ходу разговора. Вторая стратегия используется, когда аналитик имеет уже общую картину и ему требуется уточнить детали либо когда интервьюер не обладает знаниями для ответа на более общие вопросы. По мере сбора информации и прояснения общей картины можно переходить к структурированным интервью, у которых вопросы готовятся заранее, преобладают закрытые вопросы.

- Совместная разработка приложений.

Совместная разработка приложений JAD (Joint Application Development or Design) — процесс управления, который помогает ИТ-специалистам эффективнее взаимодействовать с пользователями, в целях разработки информационно-технологических решений, которые действительно работают [1].

Концепция совместной разработки приложений основана на четырех простых идеях:

- ✓ люди, фактически выполняющие работу, имеют лучшее понимание этой работы;
- ✓ люди, обученные информационной технологии, имеют лучшее понимание возможностей этой технологии;
- ✓ информационные системы и бизнес-процессы редко существуют в изоляции — они преступают формальные границы и работают в связанных отделах. Люди, работающие в этих связанных сферах, имеют ценное понимание роли, выполняемой системой в большом сообществе;
- ✓ лучшие информационные системы разработаны, когда все эти группы сотрудничают на проекте как равные партнеры.

Совместная разработка приложений представляет собой семинар с участием модератора, использующийся в области разработки программного обеспечения. Такой подход предоставляет пользователям возможность встретиться с командой разработчиков для оптимизации процесса разработки новой системы. Основная задача модератора — организовать общение между заказчиком и клиентом так, чтобы в ходе семинара постараться выявить требования к информационной системе. Роль разработчика в процессе совещания в основном сводится к пассивному слушанию для наилучшего понимания предметной области.

Совместные семинары позволяют не только непосредственно пообщаться с пользователями, но и дают дополнительные преимущества за счет командной работы. Такой подход способствует сокращению количества ошибок и недопонимания. К недостаткам этого метода можно отнести существенные затраты.

- Анкетирование.

Анкетирование — способ извлечения информации, предполагающий письменные ответы пользователей на заранее подготовленные вопросы. Этот метод обычно применяется в дополнение к другим методам. В качестве недостатка анкетирования можно отметить тот факт, что не всегда пользователи способны или хотят хорошо и информативно ответить на вопросы анкеты. Достоинство метода заключается в том, что для подготовки и анализа анкет требуются небольшие ресурсы.

Рекомендуется формулировать в анкетах вопросы с замкнутым циклом ответов. Различают три формы вопросов [9]:

- ✓ многоальтернативные вопросы — вопросы, которые имеют заранее сформулированные варианты ответов, иногда допускаются ответы в свободной форме;
- ✓ рейтинговые вопросы — вопросы, которые представляют определенный набор ответов на сформулированные вопросы, используются такие варианты ответов, как «абсолютно согласен», «согласен», «отношусь нейтрально», «не согласен», «абсолютно не согласен», «не знаю»;
- ✓ вопросы с ранжированием — вопросы, которые предусматривают упорядочивание ответов путем присваивания им порядковых номеров, процентных значений и т. п.

- Анализ документов.

Данный метод включает в себя изучение должностных инструкций, описание существующих информационных систем и бизнес-процессов.

- **Наблюдение.**

Наблюдение за работой потенциальных пользователей системы — простой и полезный способ получения информации об их работе. Оно позволяет узнать, как происходит выполнение бизнес-процесса, может дополнять интервью. Существует пассивное и активное наблюдение. При активном наблюдении аналитик работает вместе с остальными участниками бизнес-процесса, что позволяет выявить различные особенности выполнения процессов на предприятии и нестандартные ситуации.

Следует отметить, что нахождение стороннего наблюдателя может привести к тому, что сотрудники будут работать не так, как обычно, более «правильно».

Выше были представлены методы, которые позволяют получить знания о предметной области. Теперь рассмотрим некоторые методы, которые помогут аналитику заставить пользователей критически рассматривать информацию о моделях «как есть» и «как будет».

3.2. Методы определения узких мест в модели «как есть»

Анализ проблем — наиболее простой и часто используемый метод. Анализ проблем заключается в том, что специалистов предметной области просят определить проблемы в модели «как есть» и описать, как их решать в будущей системе. Большинство пользователей предлагают интересные идеи. Следует отметить, что полученные в результате этого изменения, как правило, небольшие (добавить поле, отчет и т. п.), но эффективно улучшат возможности системы.

В результате анализа проблем появляются предположения об их причинах, которые могут или не могут быть верными. Опыт разработки показывает, что обычно пользователи не склонны тщательно рассматривать причины проблем, а быстро предлагают их решение. Иногда предлагаемое решение на самом деле устраняет симптомы проблемы, а не саму проблему — основную причину. Анализ основных причин позволяет устранить проблемы.

Анализ основных причин начинается в тот момент, когда пользователь составил список проблем. Необходимо расставить приоритеты важности проблем. Затем анализируют самую важную проблему

и находят для нее все возможные основные причины; анализируют их и находят причины для них. Таким образом получают дерево причин и находят истинные причины проблемы. Важно в анализе основных причин бросать вызов очевидному и изучать проблему достаточно глубоко, чтобы понять базовые причины.

Анализ времени выполнения бизнес-плана (БП) позволяет детально проверить длительность выполнения каждого процесса в системе «как есть». Сначала аналитик определяет общее время, которое в среднем затрачивается на выполнение БП для типичных входных данных. Затем определяет время каждого отдельного этапа или подпроцесса БП. Время для завершения основных этапов суммируется и сравнивается со временем выполнения всего БП. Значительное расхождение между ними (может быть 10–100 раз, весь процесс гораздо длиннее, чем сумма времен его этапов) означает, что эта часть процесса нуждается в преобразовании (реинжиниринге).

Такая ситуация может возникнуть, если процесс плохо разделен на части. Процесс, в котором много различных людей работает на небольших участках, является кандидатом на улучшение. Возможно, требуется интеграция процесса или распараллеливание. Интеграция процесса означает изменение базового процесса так, чтобы меньше людей работало со входными данными. Распараллеливание означает одновременное выполнение нескольких этапов.

Калькуляция на основе действий — анализ, похожий на анализ времени выполнения БП. В этом методе проверяется стоимость каждого главного процесса или этапа в бизнес-процессе, а не время. Анализ определяет стоимость, связанную с каждым функциональным процессом, наиболее дорогие процессы и пытается их улучшить.

Для оценки затрат необходимо проверить прямые затраты на зарплату и материалы для каждого процесса. Также можно учитывать и не прямые затраты.

Бенчмаркинг — процесс тщательного рассмотрения бизнес-процессов организаций и их производительности в целях сопоставления их деятельности с деятельностью ведущих в данной отрасли компаний и корпораций, чтобы использовать полученные данные для приближения и даже обгона лучших в едином классе предприятий [10].

Неформальный бенчмаркинг довольно близок к клиентоориентированным БП. При этом менеджер и аналитик думают о других организациях или посещают их под видом простых клиентов, чтобы посмотреть, как выполняются БП. Во многих случаях изучать бизнес можно у известных лидеров отрасли или просто у близких конкурентов.

Анализ результатов фокусируется на понимании фундаментальных результатов, которые обеспечивают ценность для клиентов. В этом методе системный аналитик поощряет менеджеров и спонсоров проекта представить себя на месте клиента и подумать о том, что клиент действительно хочет получить от сервиса или системы предприятия и что он может получить сейчас.

Очень многие изменения в бизнесе стали возможны благодаря новым технологиям. Анализ технологий начинается с составления списка важных и интересных технологий; затем систематически определяется, как технология может быть применена в БП и какие преимущества получит БП.

Аналитики и менеджеры вместе определяют, как организация может сократить каждую деятельность в БП, как функция будет работать без нее и какой эффект будет. Вначале менеджеры неохотно заключают, что процесс можно устранить. В некоторых случаях результат получится глупый, тем не менее необходимо рассмотреть все активности.

Рассмотренные выше методы позволяют проанализировать различные аспекты, поэтому нельзя сказать, что какой-то метод лучше. Следует выбирать методы в зависимости от ситуации. Анализ проблем и анализ основных причин подходят для проектов, в которых важен рост эффективности. Анализ времени и калькуляция на основе действий помогают найти узкие места в БП, чтобы затем их улучшить. В совершенно новом для команды проекте подходят анализ результатов, неформальный бенчмаркинг и анализ технологий. Эти методы позволяют шире взглянуть на разрабатываемую систему, поискать новые пути.

Анализ вариантов использования позволяет прояснить и очертить в деталях взаимодействие пользователя и системы, которые будут в новой системе. Это позволяет лучше понимать систему.

Рассмотренные выше методы позволяют собрать информацию о новой системе, требованиях к ней. Перейдем к рассмотрению методов для документирования функциональных требований путем построения различных моделей. Эти модели создаются в рамках этапа анализа, который называется определением функциональных требований. При этом помним, что различные этапы анализа делаются параллельно и на каждой итерации проекта.

Вариант использования (use case), или прецедент — это описание последовательности действий, которые может осуществлять система в ответ на внешние воздействия пользователей или других программных систем.

Создание вариантов использования позволяет лучше понять взаимодействие пользователя и сложной системы, убедиться, что существенные задачи или шаги не пропущены перед началом варианта использования или имеются какие-то особые обстоятельства.

Разработка вариантов использования может проводится как часть интервью пользователя или JAD-сессии.

Варианты использования — хороший способ для понимания требований пользователей. Однако, есть вероятность для аналитиков-новичков неправильно предполагать, что варианты использования — это все, что нужно для полного определения того, что система должна делать. Варианты использования действительно объясняют взаимодействие пользователя с системой, но они опускают много деталей, которые необходимо знать перед разработкой системы. Они передают только точку зрения пользователя. Преобразование ее в видение разработчика путем создания функциональных требований есть одно из важных действий, которые системный аналитик делает при разработке проекта.

Наиболее общий способ для определения вариантов использования — это декомпозиция событий. Для начала нужно определить все бизнес-события, на которые должна ответить информационная система. Это поможет определить варианты использования, потому что каждое такое элементарное бизнес-событие приводит к прецеденту. Элементарное бизнес-событие — такая задача, которая выполняется одним человеком в одном месте в ответ на бизнес-событие, добавляет измеряемую бизнес-ценность и оставляет систему и данные в стабильном устойчивом состоянии.

Еще один способ проверить правильность определения прецедентов — прецедент затрагивает работу с данными (создание, чтение, изменение, удаление).

Процесс формирования прецедентов состоит из следующих этапов:

- определения вариантов использования;
- определения главных этапов внутри каждого варианта использования, т. е. тех, которые необходимы для завершения задачи;
- определения для каждого этапа триггеров, входов и выходов;
- проверки варианта использования на полноту и правильность.

4. Моделирование процессов

Ранее мы ознакомились с методами выявления требований к информационной системе. Теперь рассмотрим вопрос построения модели процессов, которая показывает последовательность выполняемых функций, перемещение данных в ходе выполнения процессов. Моделирование процессов позволяет лучше понять работу организации. На практике применяют структурные методы анализа бизнес-процессов. Структурным анализом называется метод исследования системы, начинающий с ее общего обзора, который затем детализируется, приобретая иерархическую структуру со все большим числом уровней [3]. Можно выделить следующие его существенные особенности:

- разложение системы на уровни детализации описания функций с ограничением числа элементов на каждом уровне;
- включение только существенных с точки зрения моделирования деталей на каждом уровне;
- использование строгих формальных правил записи;
- конечное число итераций детализации.

В структурном анализе и проектировании используют различные модели, наиболее распространенными являются:

- функциональная модель SADT (IDEF0), которая описывает функциональную структуру системы [3];
- модель IDEF3, описывающая процессы, в которых важно понять последовательность выполнения действий и взаимозависимости между ними; модели IDEF3 часто используются для детализации функциональных блоков IDEF0 [3];
- диаграммы потоков данных DFD, которые описывает передачу информации между функциональными процессами [3].

4.1. Метод функционального моделирования SADT (IDEF0)

Метод SADT разработан Дугласом Россом в 1969 г., использовался в военной промышленности и коммерческих организациях США. Позднее этот метод был реализован в одном из стандартов этого семейства — IDEF0.

Метод SADT представляет собой совокупность правил и процедур, предназначенных для построения функциональной модели произвольного предприятия. Функциональная модель SADT отображает функциональную структуру организации, т. е. выполняемые ей процессы и связи между ними. Для построения модели используются блоки, которые представляют собой функции, и дуги; взаимодействие блоков определяет порядок выполнения функций (рис. 4.1).



Рис. 4.1. Графические элементы модели SADT

Дуги имеют четко определенное расположение и назначение:

- вход — это информация и (или) ресурсы, которые необходимы для выполнения функции; эта дуга всегда входит слева в блок;
- выход — это информация и (или) ресурсы, которые получают-ся в результате выполнения функции; эта дуга всегда выходит из блока справа;
- управление — это нормативные документы, регламентирующие выполнение функции (законы, инструкции, положение и т. п.); эта дуга всегда входит в блок сверху;
- механизм — это то, с помощью чего выполняется функция (человек, инструмент, программа и т. п.); эта дуга всегда входит в блок снизу.

На рис. 4.2 показан пример отображения в модели SADT функции составления расписания занятий сотрудниками деканата.

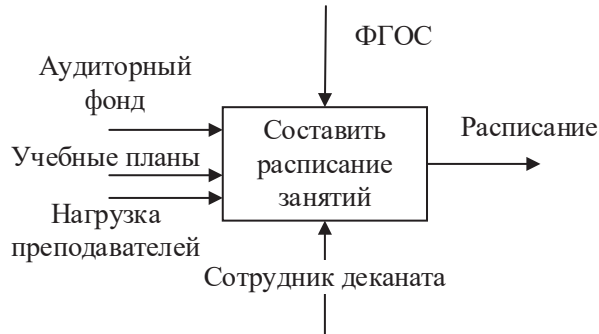


Рис. 4.2. Описание функции составления расписания в нотации IDEF0

Перед построением модели SADT необходимо определить цель построения модели и точку зрения — взгляд на систему в определенном для моделирования аспекте, поскольку это определяет набор функций, которые будут отражены в модели. Например, при построении модели процесса составления расписания, для его автоматизации будут учтены функции и сотрудники, которые непосредственно связаны с этим процессом. Следует выбрать точку зрения технолога, а не бухгалтера, потому что финансовые аспекты в данном случае не важны.

При построении функциональной модели SADT не учитывают организационную структуру предприятия. На диаграмме отображают весь процесс полностью, даже если его этапы выполняются в разных структурных подразделениях.

Построение модели SADT начинается с создания контекстной диаграммы — диаграммы верхнего уровня, которая описывает основную функцию организации и ее взаимодействие с внешней средой. Затем проводят функциональную декомпозицию, т. е. определяют, на какие функции можно разбить основную, и строят диаграмму декомпозиции. При этом следует учитывать, что дуги, входящие в блок и выходящие из него на диаграмме верхнего уровня, являются точно теми же самыми, что и дуги, которые входят в диаграмму нижнего уровня и выходят из нее. Далее функции разбивают на подфункции и т. д. В результате получается модель, которая состоит из диаграмм, комментариев и справочника терминов, которые использовались для обозначения

функций и дуг. Важный вопрос при выполнении декомпозиции — когда следует ее прекратить. Критериями останова являются:

- текущий блок содержит достаточно детальное описание выполнения функции, следующая декомпозиция начнет описывать алгоритм выполнения функции;
- для последующей детализации описания необходимо изменить точку зрения на модель, а этого делать нельзя.

По окончании построения модели следует провести критическую оценку результата — соответствуют ли описанные процессы тем, которые реально существуют на предприятии. Выявленные неточности устраняются; процесс этот итеративный и продолжается до тех пор, пока не будет принято решение о том, что полученная модель достоверна.

Методология SADT предназначена для описания высокоуровневых процессов предприятия, не позволяет описывать возможные варианты выполнения функций, поэтому при декомпозиции моделей IDEF0 иногда используют методологию IDEF3.

4.2. Методология IDEF3

Методология IDEF3 входит в семейство стандартов IDEF, она позволяет описывать технологические процессы предприятия, т.е. как конкретно выполняется определенная функция. Благодаря наличию в синтаксисе элемента «перекресток», нотация позволяет моделировать ветвления в процессе: в результате выполнения функции может выполняться сразу несколько функций или одна из них, для начала выполнения функции необходимо, чтобы завершились все или несколько предыдущих функций. Так же как и в методологии IDEF0, важно понимать цель моделирования, т.е. ответы на какие вопросы необходимо получить.

Нотация IDEF3 легка для понимания даже обычных пользователей. Она состоит из следующих элементов:

- функциональный элемент описывает действие, отображается в виде прямоугольника (рис. 4.3);
- связь описывает взаимоотношение функциональных элементов, отображается в виде стрелки;

- соединение описывает логику движения потоков между функциональными элементами.

Для наименования действий рекомендуется использовать глагол и отглагольные существительные, например: «составить расписание занятий» или «составление расписания занятий». Все действия на диаграмме нумеруются следующим образом: номер родительского действия.номер самого действия (рис. 4.3).

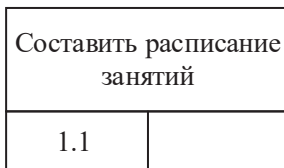


Рис. 4.3. Действие и его нумерация на диаграмме IDEF3

Все связи в IDEF3 являются однонаправленными. В отличие от методологии IDEF0, стрелка может начинаться или заканчиваться на любой стороне прямоугольника, но, как правило, стрелка входит в прямоугольник справа, а выходит — слева. Существует три возможных типа связей.

- Связь типа «временное предшествование» означает, что исходное действие должно полностью завершиться до начала выполнения следующего действия. Название связи должно содержать описание причины ее появления, можно связь не именовать (рис. 4.4).



Рис. 4.4. Пример связи «временное предшествование»

- Связь «объектный поток» показывает, что некоторый объект, полученный в результате выполнения исходного действия, необходим для выполнения последующего действия. Наименование такой связи должно содержать название объекта (рис. 4.5).

- Связь «нечеткое отношение» (рис. 4.6). Значение такой связи определяется автором диаграммы. Обычно такую связь используют

для отображения взаимоотношений между параллельно выполняющимися действиями.

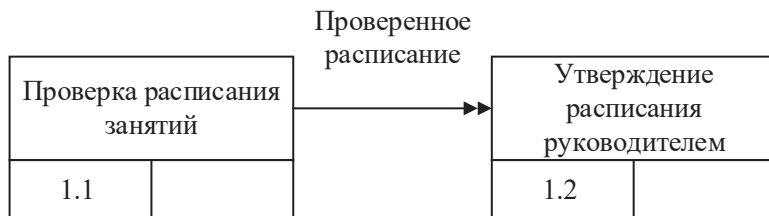


Рис. 4.5. Пример связи «объектный поток»

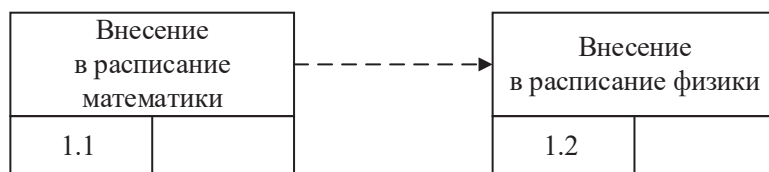


Рис. 4.6. Пример связи «нечеткое отношение»

Если после завершения одного действия начинаются сразу несколько, то соединение называется разворачивающимся. Если завершение одного действия приводит к началу сразу нескольких, то соединение называется сворачивающимся. В зависимости от типа логической операции, соединения могут быть «И», «ИЛИ» и «Эксклюзивное ИЛИ».

Соединение «И» разворачивающееся означает, что каждое конечное действие обязательно начнется (рис. 4.7).

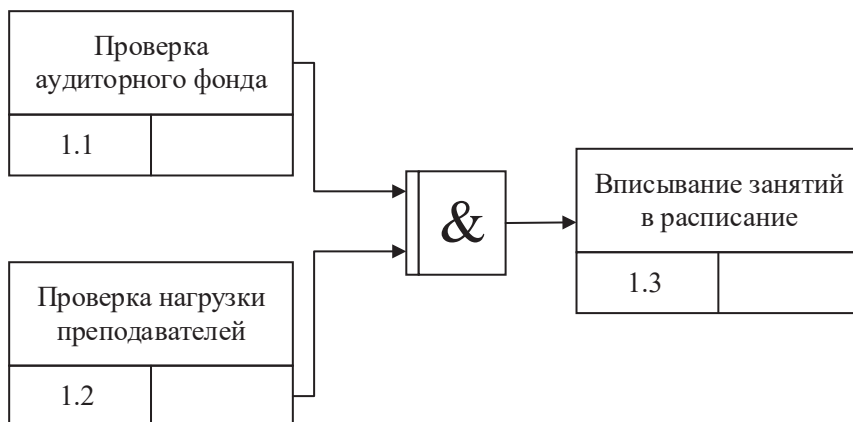


Рис. 4.7. Пример разворачивающегося соединения «И»

Соединение «И» сворачивающееся означает, что конечное действие начнется только тогда, когда завершатся все исходные действия (рис. 4.8).



Рис. 4.8. Пример сворачивающегося соединения «И»

Соединение «ИЛИ» разворачивающееся означает, что хотя бы одно из конечных действий обязательно начнется (рис. 4.9).

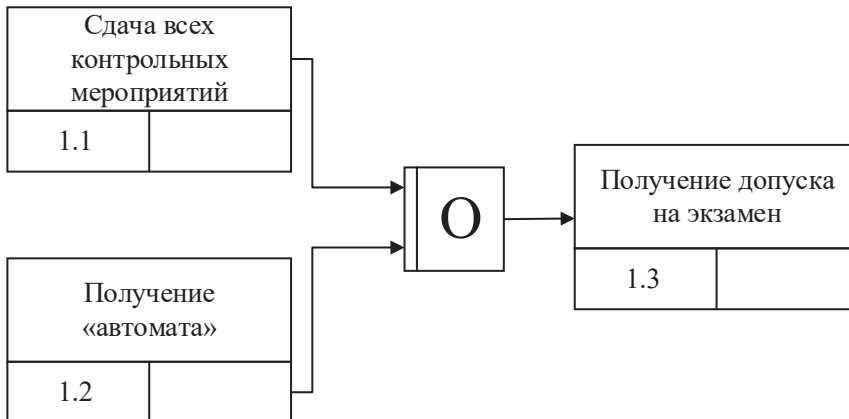


Рис. 4.9. Пример разворачивающегося соединения «ИЛИ»

Соединение «ИЛИ» сворачивающееся означает, что конечное действие начнется только тогда, когда завершится хотя бы одно из исходных действий (рис. 4.10).

Соединение «Эксклюзивное ИЛИ» разворачивающееся означает, что только одно из конечных действий обязательно начнется (рис. 4.11).

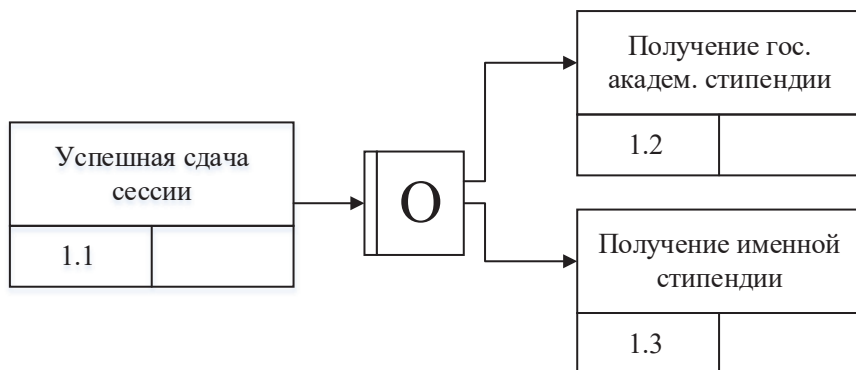


Рис. 4.10. Пример сворачивающегося соединения «ИЛИ»

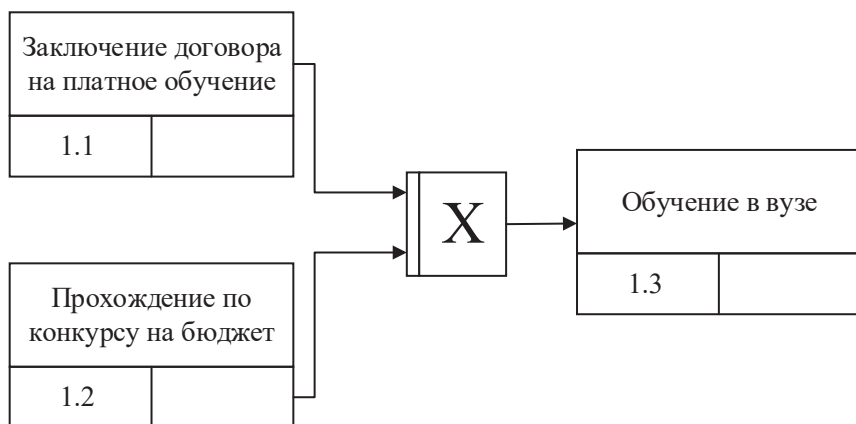


Рис. 4.11. Пример разворачивающегося соединения «Эксклюзивное ИЛИ»

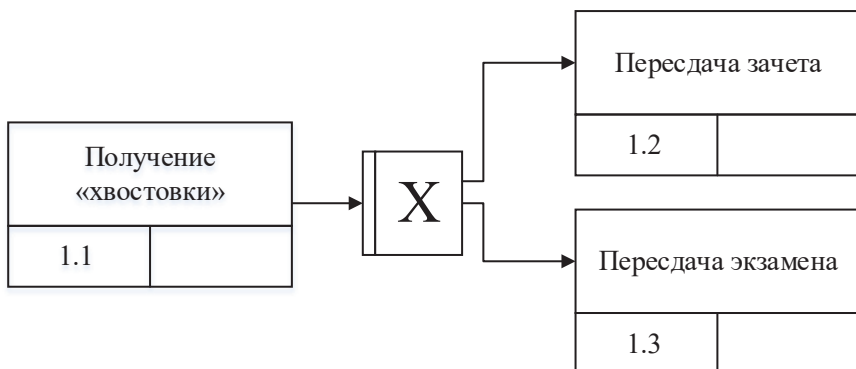


Рис. 4.12. Пример сворачивающегося соединения «Эксклюзивное ИЛИ»

Соединение «Эксклюзивное ИЛИ» сворачивающееся означает, что конечное действие начнется только тогда, когда завершится только одно из исходных действий (рис. 4.12).

Методология IDEF3 позволяет описать порядок выполнения функций в бизнес-процессе предприятия, логику взаимодействия потоков данных. Для более полного описания обработки информации при выполнении бизнес-процессов используется методология DFD.

4.3. Методология DFD

Методология DFD (Data Flow Diagrams — диаграммы потоков данных) предназначена для описания бизнес-процессов предприятия с точки зрения ввода, обработки и хранения информации. В основе этой методологии также лежит декомпозиция, а особенностью ее является наличие хранилищ данных и внешних по отношению к рассматриваемым процессам источников и приемников данных — внешних сущностей.

DFD-диаграмма — это сеть связанных между собой работ, которые преобразовывают входные данные в выходные. Она включает в себя:

- функцию — процесс, который преобразует поток данных;
- поток данных — объекты, над которыми выполняются функции: материальные объекты или информация;
- хранилище данных — внутренние хранилища данных системы, могут быть таблицами базы данных, файлами или каким-то физическим хранилищем, например сейфом, архивом или складом;
- внешняя сущность — объект, который находится за пределами рассматриваемой системы и обменивается с ней данными, но в их обработке не участвует.

Существует две нотации для построения DFD-диаграмм: Гейна — Сарсона и Йордана — Де Марко [3]. В работе используется первая нотация.

Функция на DFD-диаграмме (рис. 4.13) обозначается прямоугольником; наименование должно содержать глагол, определяющий функцию и объект, над которым это действие осуществляется. Каждая функция должна иметь уникальный номер для ссылок на него внутри диаграммы. Совместное использование этого номера с номером диа-

граммы дает уникальный индекс функции во всей модели. Поскольку функция занимается обработкой данных, то она должна иметь хотя бы один вход и один выход.

Поток данных изображается на DFD-диаграмме стрелкой, ориентация которой указывает направление движения данных. В названии потока должно присутствовать название объекта, которого он отображает, и его определение, например: «утвержденное расписание».

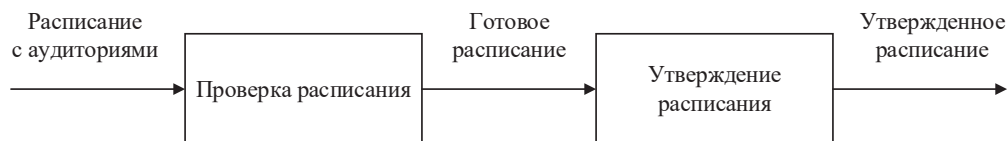


Рис. 4.13. Фрагмент DFD-диаграммы

Хранилище данных обозначается фигурой, показанной на рис. 4.14, название соответствует реальному хранилищу.

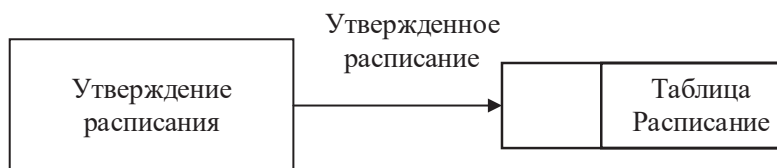


Рис. 4.14. Фрагмент DFD-диаграммы с хранилищем данных

Внешняя сущность изображается в виде прямоугольника с тенью (рис. 4.15). Одна внешняя сущность может быть использована многократно на одной или нескольких диаграммах.

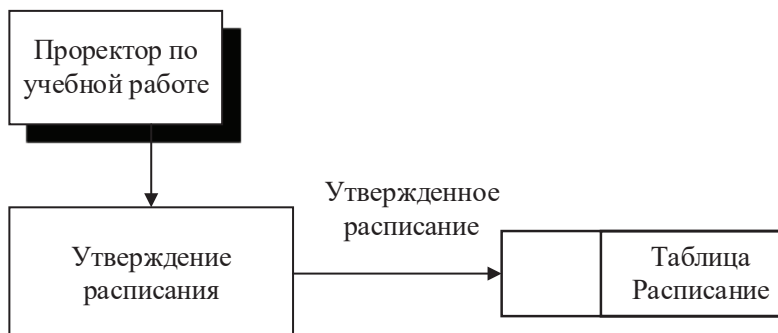


Рис. 4.15. Фрагмент DFD-диаграммы с внешней сущностью и хранилищем данных

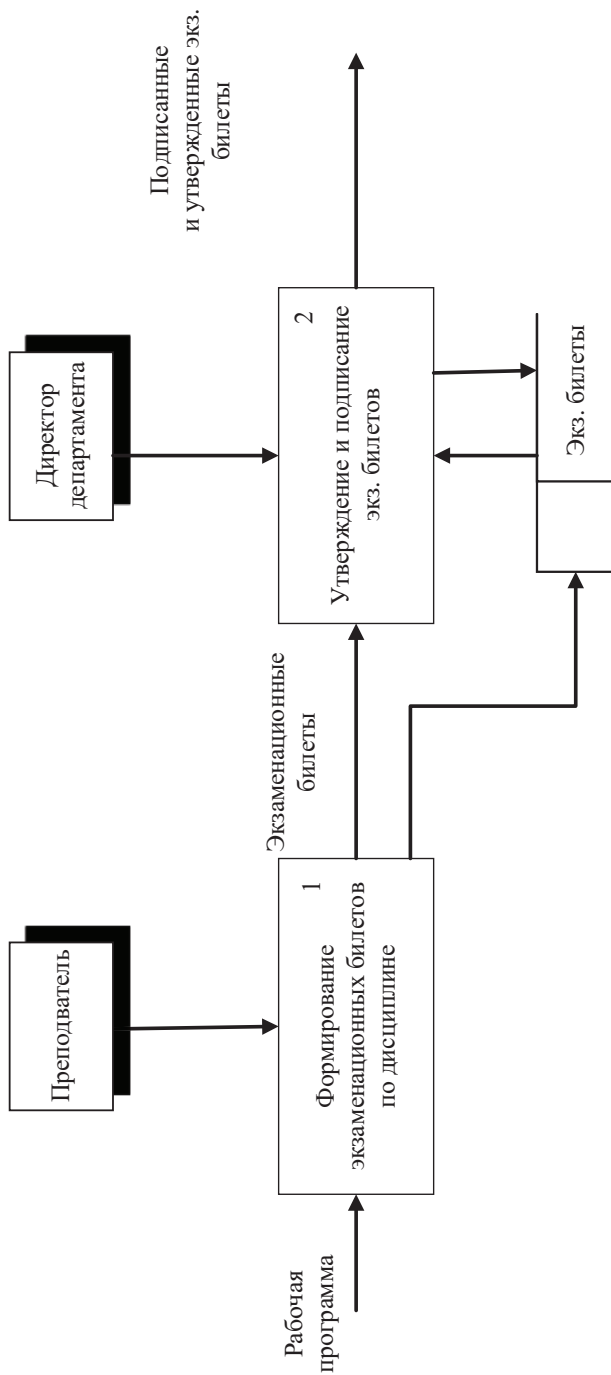


Рис. 4.16. DFD-диаграмма бизнес-процесса «Подготовка экзаменационных билетов» в нотации Гейна — Сарсона

На рис. 4.16 приведен пример DFD-диаграммы бизнес-процесса «Подготовка экзаменационных билетов», разработанной в нотации Гейна — Сарсона.

Построение иерархии диаграмм потоков данных начинается с контекстной диаграммы, на которой отображается взаимодействие системы, представленной главным процессом, с внешней средой, которая представлена одной или несколькими внешними сущностями. Для проверки контекстной диаграммы готовится список возможных действий внешних сущностей и соответствующих реакций системы на них. Если система достаточно сложная — много внешних сущностей или существует сложившийся набор подсистем с определенными функциями, то строится иерархия контекстных диаграмм. При этом контекстная диаграмма верхнего уровня содержит набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют подсистемы.

После завершения моделирования и проверки получившейся модели можно провести количественный анализ диаграмм IDEF0 и DFD [3]. Для его проведения используются следующие показатели:

- количество блоков на диаграмме j — N_j ;
- уровень декомпозиции диаграммы j — L_j ;

В литературе [3] дается следующая рекомендация по значениям этих показателей: количество блоков на диаграммах нижних уровней должно быть меньше количества блоков на родительских диаграммах, т. е. с увеличением уровня декомпозиции коэффициент N_j/L_j должен убывать.

4.4. Нотация BPMN

Рассмотренные выше методологии позволяют построить модель разрабатываемой информационной системы, а затем прийти к его проектированию и разработке. Естественно, в дальнейшем в полученной модели возможны различные изменения, которые не всегда отражаются непосредственно в модели, а содержатся только в реализованной системе. Таким образом появляется разрыв между моделью системы и ее реализацией.

В начале XXI в. возникает новый подход к управлению бизнес-процессами организации, который ориентирован на непрерывное их усо-

вершенствование. Появилась потребность в новой нотации BPMN (Business Process Model and Notation), которая позволяла бы не только моделировать процессы, но и автоматически их исполнять и контролировать. Она тесно связана с системой управления, в которых производится моделирование, а также с исполнением бизнес-процессов. Такая система состоит из трех частей: средства моделирования, средства исполнения и средства мониторинга.

Нотация BPMN состоит из большого количества элементов [11]. Ниже кратко рассмотрим только основные элементы:

- Event — Событие;
- Activity — действие;
- Gateway — шлюз или развилка;
- Flow — поток;
- Date — данные;
- Artefact — артефакт;
- Swimline — дорожка;
- Pool — пул.

Пул используется для обозначения границ бизнес-процесса, он делится на дорожки. Каждая дорожка определяет одного участника процесса. Бизнес-процесс начинается со стартового события и заканчивается конечным событием, на диаграмме обозначается в виде круга. В процессе его выполнения могут возникать промежуточные события, например «Отправка письма».

Действия — это простые функции, из которых складывается бизнес-процесс, на диаграмме обозначаются прямоугольниками.

Поток определяет последовательность выполнения действий, обозначается стрелкой. Также в нотации присутствуют пунктирные стрелки. Они обозначают сообщения, которыми обмениваются участники бизнес-процесса.

Для обозначения условной логики выполнения процесса используются шлюзы, которые графически изображаются в виде ромба.

Артефакт — это объект, который не влияет на исполнение бизнес-процесса напрямую, но поясняет его, например документ или уточняющий комментарий.

5. Этап проектирования информационной системы

На этапе анализа мы выяснили, в какой ИС нуждается бизнес. На этапе проектирования необходимо определить, как эту систему создать. Также на данном этапе определяется стратегия получения новой ИС. Система может разрабатываться с нуля самостоятельно или внешними программистами, куплена и адаптирована под требования предприятия. Необходимо проанализировать достоинства и недостатки каждого из способов и выбрать наилучший. Это решение оказывает влияние на весь этап проектирования.

В начале этапа проектирования проводится анализ нефункциональных требований — это требования, которые описывают условия работы информационной системы и ее качества. К ним относят, например:

- ограничения — условия работы ИС, которые ограничивают выбор возможных вариантов реализации компонент системы;
- правила, законы и стандарты — определяют особенности выполнения бизнес-процессов;
- внешние интерфейсы — способы взаимодействия с другими системами;
- производительность;
- надежность и устойчивость к сбоям;
- требования к безопасности.

Нефункциональные требования преобразуются в системные требования, которые определяют архитектуру системы. Она в свою очередь определяет требования к аппаратному и программному обеспечению системы. Эти требования формулируются в виде спецификации аппаратного и программного обеспечения — список программного обеспечения и оборудования, необходимого для работы ИС.

На этапе проектирования определяются способы взаимодействия пользователя с системой, логическая модель данных преобразуется в структуру базы данных. Во время проектирования на основе логи-

ческой модели процессов, разработанной на этапе анализа, создается модель физических процессов, чтобы показать детали реализации и объяснить, как будет работать конечная система. Эти детали могут включать ссылки на реальную технологию, формат передаваемых данных и участие пользователя при выполнении определенных функций.

Таким образом, этап проектирования включает в себя:

- разработку архитектуры системы;
- построение модели системных процессов;
- построение модели данных приложения;
- построение модели пользовательского интерфейса.

Этап проектирования завершается разработкой технического проекта ИС [1].

5.1. Дизайн архитектуры

Важной частью этапа проектирования является проектирование архитектуры ИС [1].

Все ИС можно разделить на четыре базовые части. Первая — это хранение данных. В зависимости от типа, данные могут храниться в файлах, реляционной или нереляционной базах данных. Второе — это логика доступа к данным. При использовании реляционной базы данных логика доступа к данным представляет собой запросы на языке структурированных запросов SQL. Третье — это логика приложения, описанная в одной из нотаций (см. гл. 4) и функциональных требованиях. Четвертое — логика представления: отображение информации для пользователя и принятие пользовательских команд. С этими частями ИС напрямую связаны три основные составляющие аппаратного обеспечения: клиентские компьютеры, серверы и сетевое оборудование. Требования к частям информационной системы определяют ее архитектуру. Также на данном этапе определяют требования к аппаратному и программному обеспечению и сетевому окружению.

Далее кратко рассмотрим наиболее часто используемые архитектуры информационных систем и их особенности.

- Классическая клиент-серверная архитектура.

Клиент-серверная архитектура делит весь процесс на две части: клиент отвечает за логическое представление, а сервер — за хране-

ние и доступ к данным. Логика приложения может быть расположена на клиенте (толстый клиент), на сервере (тонкий клиент) или разделена между ними.

Преимущества клиент-серверной архитектуры:

- ✓ масштабируемость — легко увеличить объем базы данных и скорость обработки, добавляя новые серверы;
- ✓ защищенность данных — данные хранятся на сервере, который реализует резервное копирование и разграничение прав доступа;
- ✓ низкие требования к клиентским машинам;
- ✓ надежность системы — сбой одной клиентской машины не приводит к краху системы, а серверы имеют надежность на порядок выше.

Недостатки ее:

- ✓ дорогое специальное программное обеспечение;
- ✓ сложность разработки и сопровождения — нужно поддерживать серверную и клиентскую части.
- Трехуровневая клиент-серверная архитектура.

Трехуровневая клиент-серверная архитектура разделяет приложение на три части:

- ✓ клиентская — отвечает за логику представления данных;
- ✓ сервер приложений — отвечает за логику приложения;
- ✓ сервер базы данных — отвечает за логику доступа к данным и их хранение.

Благодаря такому распределению, требования к аппаратным средствам клиента снижаются еще больше. Конечный пользователь для отображения данных использует веб-браузер. Такую архитектуру называют тонким клиентом, а рассмотренную выше классическую архитектуру — толстым клиентом.

Существенно упрощается сопровождение информационной системы, поскольку необходимо обновлять части системы на серверах приложений и базы данных.

Поскольку данная архитектура является развитием классической клиент-серверной, постольку она обладает всеми ее достоинствами. Разработка и развертывание системы с такой архитектурой усложняется. Дополнительно можно указать следующие положительные моменты архитектуры:

- ✓ низкие требования к пропускной способности сети между клиентами и сервером приложений;
- ✓ гибкость системы — благодаря разделению на сервер приложения и сервер баз данных, легче менять конфигурацию системы.
- Архитектура веб-приложения.

Веб-приложение можно разделить на следующие части: клиентскую, веб-сервер, сервер приложений и сервер базы данных. Клиентская часть представляет собой набор страниц, отображаемых в веб-браузере. Эти страницы готовит веб-сервер на основании запросов клиентской части. Сервер приложений отвечает за выполнение бизнес-логики. Отличительной особенностью этой архитектуры является отсутствие постоянного соединения между клиентом и веб-сервером. Это создает определенные проблемы при разработке, которые решаются при помощи использования сессий — промежутка времени, в течение которого веб-приложение может определять все запросы от одного клиента.

- Сервис-ориентированная архитектура.

Сервис-ориентированная архитектура — это подход, при котором необходимая бизнес-логика реализуется в виде некоторого количества программных модулей (сервисов), взаимодействующих друг с другом и пользователями с помощью стандартизованных интерфейсов по стандартизованным протоколам. Такая архитектура обеспечивает более быструю и простую интеграцию приложений компании.

При выборе архитектуры системы учитывают:

- ✓ существующую инфраструктуру компании;
- ✓ корпоративные стандарты, лицензионные соглашения;
- ✓ опыт работы проектной команды.

Если у предприятия есть несколько инфраструктур, то можно провести тестирование и выбрать наиболее подходящую для задачи.

Существует четыре основных типа нефункциональных требований, которые могут быть важны при проектировании архитектуры: эксплуатационные требования, требования к производительности, требования безопасности, а также культурные и политические требования. Далее кратко рассмотрим каждое из них.

5.2. Эксплуатационные требования

Эксплуатационные требования определяют среду, в которой предстоит работать информационной системе [1]. Под средой понимают:

системное программное обеспечение; ИС, с которыми необходимо взаимодействовать; а также особенности физической среды, если они оказывают влияние на работу системы (шум, жара, влажность и т. п.). Эксплуатационные требования делятся на требования:

- к технической среде — тип аппаратного и программного обеспечения, на котором будет работать ИС, например операционное система, СУБД;
- к системной интеграции — определяют ИС, внешние или внутренние, с которыми необходимо взаимодействовать, а следовательно, интерфейс обмена данными;
- к переносимости — определяют, как требования к технической среде могут изменяться с течением времени и как на эти изменения должна отвечать система (например, новая ОС). Они также ссылаются на потенциальные изменения в бизнес-требованиях, которые окажут влияние на технические требования (новое оборудование пользователей — смартфоны);
- эксплуатации и техническому обслуживанию — определяют режим работы информационной системы (круглосуточно или только в рабочее время), квалификацию и количество персонала, которые необходимы для обслуживания работы системы, регламент обслуживания.

Анализ эксплуатационных требований, предъявляемых к системе, может обосновать преимущество одной архитектуры над другими. Например, если система должна интегрироваться с приложениями, установленными на компьютере пользователя, то это не сильно ограничивает в выборе архитектуры. Для системы с обширными требованиями к переносимости, как правило, лучше всего подходит клиент-серверная архитектура с тонким клиентом.

5.3. Требования к производительности

Требования к производительности определяют такие параметры, как время ответа системы, максимальное количество одновременно работающих пользователей, надежность [1]. Каждое требование должно быть измеряемым, только в этом случае требования к производительности могут быть проверены. Хорошей практикой является при

включении требований к производительности в техническую документацию сразу же описывать четкие условия проведения тестирования для каждого требования. Такой подход предотвращает создание плохо сформулированных требований к производительности, которые невозможно проверить. Например, время ответа системы должно быть достаточным для того, чтобы персонал мог эффективно завершить свою работу. Это пример нетестопригодного требования. Правильная формулировка — время отклика системы на запрос пользователя должно составлять не более 4 с. Более подробно опишем требования к производительности.

- Требования к скорости определяют скорость работы системы. Работа системы может рассматриваться в разных аспектах, прежде всего это время ответа системы. Очевидно, что обеспечить мгновенность на все действия пользователя невозможно. Следует различать ответ клиентской части и серверной. Реакция системы на работу с компонентами пользовательского интерфейса должна быть в пределах нескольких секунд. Время выполнения операций или обработка запросов зависит от их сложности. Другой аспект — это как быстро транзакция в одной части системы окажет влияние на другие части, например система заказов. Один пользователь купил определенный товар — насколько быстро у других пользователей количество этого товара уменьшится.
- Ограничения системы устанавливают предельные значения различных показателей, например: как много пользователей система будет поддерживать в целом и одновременно, максимальный объем хранимых данных. Это позволяет понять размер БД и необходимые вычислительные мощности. Самое важное требование — максимальное количество одновременно работающих пользователей, потому что оно оказывает прямое влияние на производительность компьютеров, которые требуются для работы системы.
- Требования к надежности определяют время восстановления работы системы после отказов и сбоев. Обычно надежность обеспечивается за счет резервирования серверов и резервного копирования программ и данных по установленному графику.

Требования к производительности больше влияют не на выбор архитектуры системы, а на выбор аппаратного и программного обеспечения.

5.4. Требования к безопасности

Безопасность — это способность защитить ИС от сбоев и потери данных, которые вызваны умышленными действиями или случайными событиями [1]. Безопасность системы во много определяется тем, как она установлена и эксплуатируется, какие средства контроля безопасности при этом используются. Тем не менее разработчики информационных систем также должны гарантировать, что ими приняты разумные меры по обеспечению безопасности. Для этого они прежде всего должны ознакомиться с требованиями по информационной безопасности, которые применяются для различных организаций на законодательном уровне.

Перед формулировкой требований к безопасности необходимо определить цену уязвимости в защите данных ИС. Очевидно, что серьезные меры защиты, предпринимаемые для обеспечения информационной безопасности, требуют больших финансовых затрат. Можно использовать следующее эмпирическое правило: оценить сумму ожидаемой потери и вероятность ее возникновения. Умножение этих значений дает оценку того, сколько нужно потратить на меры безопасности. Таким образом, ожидаемый убыток, например, в размере 100 млн р. с вероятностью 10 % означает, что на безопасность следует потратить до 10 млн р.

Перед разработкой системы необходимо определить требования к контролю доступа, которые определяют, какие группы пользователей могут получить доступ к каким данным и с какими правами (чтение, изменение, удаление). Также должна быть определена необходимость в криптографической защите данных, использовании электронно-цифровой подписи и способе аутентификации.

Требования к безопасности косвенно влияют на выбор архитектуры системы. Они обеспечиваются в результате разработки мер при ее проектировании и эксплуатации, направленных на защиту информационной системы от сбоев и потери данных.

5.5. Культурные и политические требования

Широкое использование Интернета в бизнесе привело к тому, что система, разработанная для российских клиентов, может быть доступна и за ее пределами. Поэтому важной частью проектирования архитектуры системы является понимание глобальных культурных и политических требований к системе. Кратко рассмотрим эти требования [1].

- Поддержка нескольких языков — наиболее очевидное требование для ИС, которую будут использовать в разных странах. В связи с этим возникают проблемы перевода специфических терминов на иностранный язык. Другой проблемой является распределение пространства экрана. На разных языках фразы имеют разную длину, поэтому занимают разное место на экране. Следующая проблема — способ реализации многоязычия. Можно менять язык интерфейса налету, а можно реализовать выбор языка при установке системы. Все эти особенности следует описывать на этапе проектирования информационной системы.
- Индивидуальные настройки позволяют конкретным пользователям изменять интерфейс системы и, возможно, правила или порядок выполнения определенных функций. Такие системы являются более гибкими с точки зрения потребностей клиентов. Однако это усложняет разработку и поддержку системы.
- Настройка страновых особенностей. Под страновыми особенностями понимаются формат даты, валюта, мера весов. При разработке глобальных систем очень важно учитывать такие особенности каждой страны и предоставлять пользователю информацию в привычной для него форме.
- Правовые требования устанавливаются законами и государственными нормативными актами. Разработчики системы иногда забывают учитывать их, а это может привести к серьезным негативным последствиям.

Реализация культурных и политических требований существенно упрощается при разделении логики приложения и логики обработки данных. Следовательно, эти требования напрямую влияют на выбор архитектуры информационной системы.

6. Разработка пользовательского интерфейса

Пользовательский интерфейс (ПИ) — это часть ИС, которая отвечает за взаимодействие с пользователем. Следует отметить, что кроме ПИ есть еще и системные интерфейсы, с помощью которых новая ИС обменивается данными с другими ИС. Эти интерфейсы разрабатываются как часть архитектуры системы на основе требований интеграции.

ПИ включает в себя три части:

- механизм навигации — способ передачи инструкции пользователя ИС для выполнения каких-либо действий, например меню, панель инструментов;
- механизм ввода данных — способ, с помощью которого ИС принимает информацию, например форма ручного ввода данных;
- механизм вывода данных — способ, с помощью которого, ИС предоставляет данные пользователям, например бумажный отчет.

Разрабатывая навигацию ИС, аналитик должен учитывать особенность обычного пользователя, который не будет изучать руководство пользователя, проходить обучение или обращаться к системе помощи. Структура навигации и названия ее элементов должны быть понятны пользователю сразу. Пункты меню и кнопки, которые в данный момент не могут быть использованы, должны быть недоступны пользователю. Критические операции, например удаление чего-то, должны сопровождаться сообщением с подтверждением.

Проектирование механизма ввода данных — это проектирование экранных форм для ввода информации в систему. При их создании следует максимально учитывать вид источников этих входных данных в реальной жизни. Например, поля формы должны в точности повторять порядок и расположение полей в бумажном бланке и иметь такие же названия. Исходя из условий эксплуатации системы следует использовать онлайн и пакетную обработку данных (сразу сохранять введенные данные или нет).

При проектировании механизма вывода данных следует учитывать то, как используются результаты, какое их представление будет проще и понятней для пользователей.

Кратко ознакомимся с базовыми принципами разработки ПИ [12].

6.1. Базовые принципы разработки ПИ

При разработке ПИ следует учитывать следующие базовые принципы [1]:

- расположение — форму или веб-страницу необходимо разделить на несколько областей и использовать каждую из них всегда для одних и тех же целей. Например, вверху — меню, панели инструментов, в середине — область для ввода-вывода информации, внизу — область для информации о состояниях системы;
- понимание текущего содержания — пользователь всегда должен понимать, где он находится в системе и что за информация перед ним представлена. Для этого используются понятные подписи всех элементов ПИ, всплывающие подсказки;
- эстетика — ПИ должен быть не только функциональным, но и привлекательным для пользователя. Не следует перегружать форму или веб-страницу системы, надо грамотно использовать цвета и разные шрифты. Следует располагать темное содержимое переднего плана на светлом фоне или наоборот. Не надо использовать красный и зеленый цвета для контрастной окраски элементов интерфейса, т. к. многие люди, страдающие дальтонизмом, просто не смогут увидеть разницы. Решение интерфейса в холодных цветах подходит для строгих офисных приложений;
- опыт пользователей — при разработке ПИ необходимо учитывать привычки пользователей, которые вырабатываются у них при использовании аналогичных приложений. Например, сочетание клавиш <Ctrl> + <C> и <Ctrl> + <V> во многих приложениях позволяет скопировать и вставить текст. Разумно выполнение аналогичных функций в своем приложении запрограммировать через эти же клавиши;
- последовательность (логичность) — этот принцип построения ПИ позволяет пользователям предугадать следующие действия для вы-

полнения нужной им операции. Для этого необходимо хорошо изучить порядок выполнения бизнес-процессов пользователя;

- минимизация усилий пользователя — ПИ должен быть прост в использовании, функции должны выполняться за минимальное количество действий.

Процесс проектирования пользовательского интерфейса состоит из пяти этапов, которые выполняются итеративно:

- разработки пользовательских сценариев — описание часто используемых действий пользователей и способы их выполнения с помощью ПИ, чтобы позднее пользователи могли быстро и без проблем выполнять эти сценарии;
- разработки структурных диаграмм ПИ, которые описывают базовые элементы ПИ и их взаимодействие при работе пользователей;
- разработки стандартов ПИ — соглашения по наименованию элементов ПИ и их размещению;
- разработки прототипа ПИ — модели системы, которая демонстрирует основные принципы будущего ПИ системы;
- оценивания ПИ — получение обратной связи от пользователей вплоть до завершения всего проекта.

Какой-то определенной нотации для структурных диаграмм интерфейса нет. Обычно диаграммы состоят из прямоугольников — элементов ПИ и линий — путей навигации между ними (рис. 6.1). В блоках пишут названия элементов ПИ и номер. Нумеровать блоки можно по-разному. Диаграммы обычно имеют древовидную структуру. Базовая структура ПИ следует из базовой структуры БП. Аналитик начинает с DFD и разрабатывает основной поток управления системой, двигаясь от процесса к процессу. Обычно существует несколько независимых главных частей ИС, которые соответствуют нескольким БП верхнего уровня.

После построения диаграммы аналитик проверяет, насколько диаграмма соответствует сценарию использования. Очень часто оказывается, что для реализации сценария необходимо пройти достаточно сложный путь по диаграмме. В этом случае аналитику необходимо переделать диаграмму, добавляя короткие переходы между элементами, например, меняя структуру меню, добавляя комбинации горячих клавиш, чтобы реализовать пользовательский сценарий наиболее простым способом.

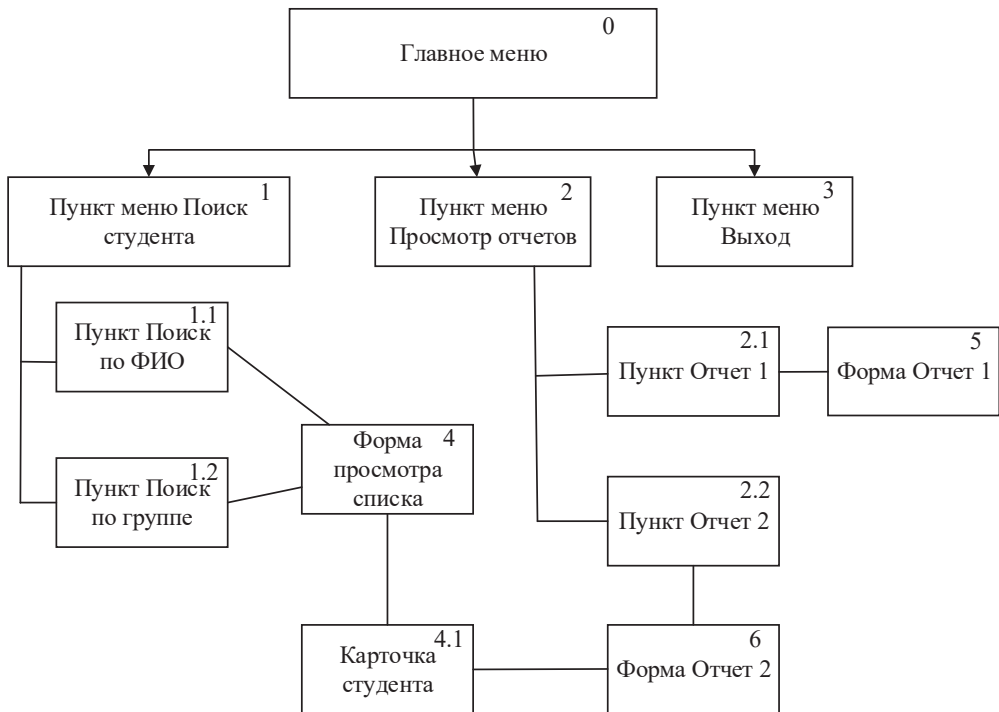


Рис. 6.1. Пример структурной диаграммы интерфейса

Важной составляющей разработки ПИ является создание стандартов ПИ — основных элементов дизайна, которые являются общими для отдельных экранов, форм и отчетов в системе. В зависимости от типа приложения может быть несколько наборов стандартов ПИ для различных частей системы (web-формы, отчеты, формы ввода информации). Благодаря стандарту ПИ все части системы, разработанные командой программистов, выглядят единообразно. Рассмотрим, что же входит в понятие «стандарт ПИ».

- **Метафоры.**

Метафора — сущность, которая заменяет объект реального мира в информационной системе. Они помогают пользователю понять систему и дают ему возможность предугадать, что делает тот или иной элемент ПИ. Самая известная метафора — это рабочий стол операционной системы Windows.

В некоторых случаях метафору очень легко придумать. Например, для интернет-магазинов метафорой являются обычные магазины, поэтому «корзина» и «кошелек» выглядят очень естественно. В других

случаях метафору трудно определить. В целом лучше не использовать метафору, которая на самом деле не соответствует системе, потому что неверная метафора только запутает пользователя.

- **Шаблоны.**

Шаблоны определяют общий вид всех экранных форм ИС и отчетов. Они содержат цветовую схему, стандартное размещение строк состояния системы, порядок для общих действий интерфейса. Например, кнопка вызова контекстной помощи всегда находится в правом верхнем углу формы.

- **Словарь терминов.**

Словарь терминов содержит название команд и объектов предметной области. Например, кнопка закрытия формы везде называется «Выход», а не «Закрыть» или «Выйти».

- **Иконки.**

Очень часто пункты меню или кнопки имеют картинку-иконку. Некоторые действия ИС, такие, как «открыть файл», «печать», уже имеют устоявшиеся у пользователей иконки. При разработке своей системы следует использовать для традиционных команд привычные иконки, для специфических действий нужно очень аккуратно подходить к выбору этих маленьких картинок. Не всегда пользователю понятно значение этих странных значков. Хорошее решение проблемы — всплывающие подсказки с текстом, которые появляются, когда указатель мыши находится над значком, они объясняют назначение значка словами.

6.2. Использование CASE-средства BPsim.SD при проектировании пользовательского интерфейса

При проектировании ИС большой интерес представляет анализ информационных потоков, который удобно проводить с использованием методологии DFD (нотация Гейна — Сарсона). Рассмотрим технологию проектирования ПИ для функции «Регистрация входящего документа» системы ведения реестров (СВР) «ВЕРЕКОМ-2». С ее помощью пользователь регистрирует входящие документы в базе данных СВР «ВЕРЕКОМ-2». Эти документы относятся к акционерам — лицам, зарегистрированным в этой системе. DFD-диаграмма процесса

регистрации входящего документа показана на рис. 6.2. Она разработана в CASE-средстве BPsim.SD.



Рис. 6.2. DFD-диаграмма процесса регистрации входящего документа в BPsim.SD

DFD-диаграммы представляют собой результаты проведенного структурного анализа системы. В CASE-средстве BPsim.SD они являются основой для объектно ориентированного проектирования. Применение объектно ориентированного подхода позволяет повысить уровень унификации разработки и пригодность ее для повторного использования, что удешевляет проект за счет использования предыдущих разработок.

К методам объектно ориентированного проектирования относятся диаграммы прецедентов, последовательности, классов языка UML.

Диаграммы прецедентов демонстрируют взаимодействие между различными вариантами использования информационной системы и действующими лицами (пользователями) этой системы, отражая функциональные требования к системе с точки зрения пользователя.

Переход от DFD-диаграмм (см. рис. 6.2) к диаграмме прецедентов можно осуществлять в автоматическом (рис. 6.3) или полуавтоматическом режиме (рис. 6.4). Алгоритм формирования диаграммы прецедентов каждой функции на DFD-диаграмме ставит в соответствие прецедент, каждой внешней сущности (DFD) — роль (актер, действующее лицо) [13].

Полуавтоматический режим позволяет автоматизировать проектирование больших и сложных диаграмм, выделять из них функциональные модули или рассматривать взаимодействие отдельных актеров с информационной системой. Пример диаграммы прецедентов для процесса регистрации входящего документа приведен на рис. 6.3.

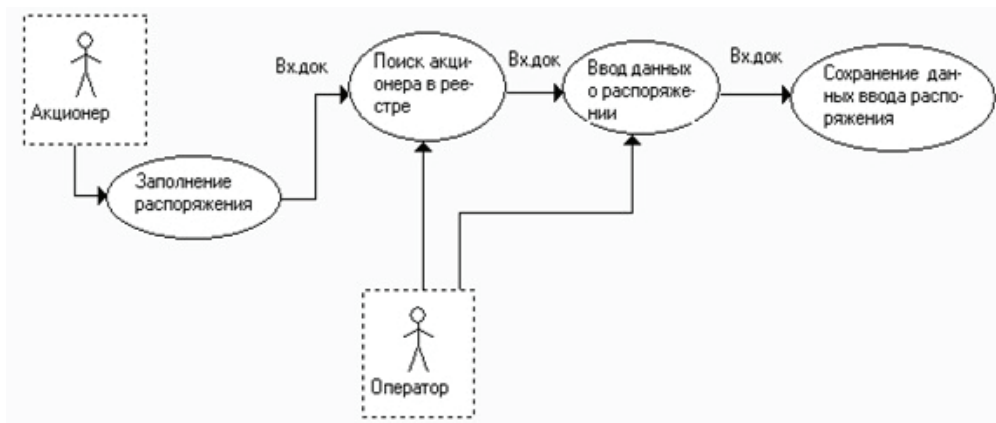


Рис. 6.3. Диаграмма прецедентов для процесса регистрации входящего документа в BPsim.SD

Диаграммы последовательности позволяют показать временную последовательность событий, которые происходят в рамках определенного варианта использования. На диаграмме показывается ряд объектов (пользователи, экземпляры программных классов) и сообщения, которые отражают взаимодействие между ними. Для каждого варианта использования диаграммы прецедентов (рис. 6.3 и 6.4) с помощью диаграмм последовательностей описывается порядок выполнения операций, смены экранных форм, взаимодействия программных объектов информационной системы и базы данных (менеджер транзакций).

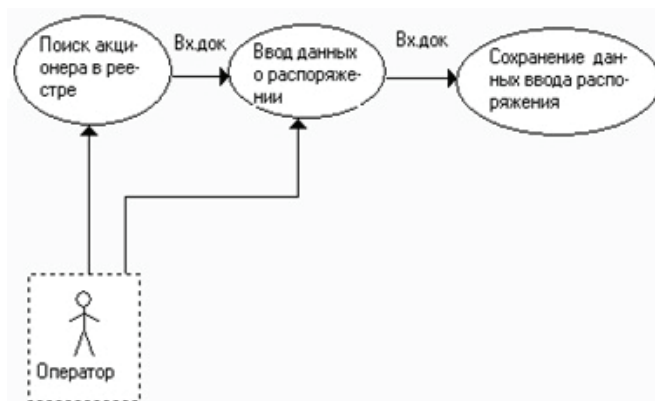


Рис. 6.4. Диаграмма прецедентов для роли «Оператор» в BPsim.SD

На рис. 6.5 приведена диаграмма последовательности процесса «Регистрация входящего документа».

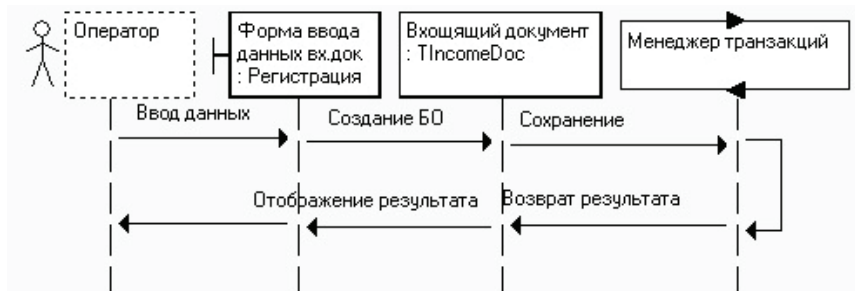


Рис. 6.5. UML-диаграмма последовательности процесса «Регистрация входящего документа» в BPsim.SD

Для каждого объекта диаграммы последовательности определяется класс, описываются атрибуты, типы, связи, которые существуют между ними, операции и ограничения, которые накладываются на связи между этими классами. При проектировании пользовательского интерфейса разрабатываемой системы, каждый объект диаграммы последовательности привязывается к одному из четырех типов пакетов:

- бизнес-объектам;
- управлению — управляющие объекты;
- границам — интерфейсы взаимодействия между подсистемами и графические интерфейсы;
- актерам.

Система проектирования пользовательского интерфейса позволяет по существующим диаграммам последовательности и классам моделировать пользовательский интерфейс разрабатываемой информационной системы. За основу берется объект — граница, содержащая список экранных форм, необходимых конкретному пользователю (или нескольким пользователям) для выполнения рассматриваемой задачи (варианта использования информационной системы).

В соответствии с типами и другими свойствами бизнес-объектов, описанных в привязанном к данному объекту классе, выбираются наиболее подходящие компоненты для отображения соответствующих данных на формах информационной системы. Разработчику пользовательского интерфейса информационной системы предлагается для

каждого атрибута бизнес-объекта выбрать из предложенных системой проектирования нужный визуальный компонент и разместить его на форме. После создания и описания форм каждого прецедента получаем проект ПИ разрабатываемой информационной системы, который впоследствии может быть изменен. В рассмотренном нами примере, для бизнес-объекта «Входящий документ» создан класс «TIncomeDoc» (рис. 6.6). Для него определены необходимые атрибуты.

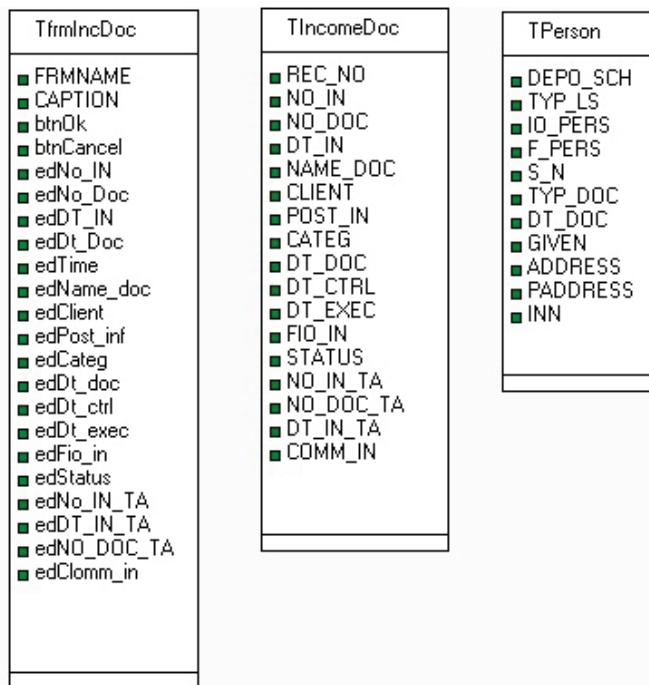
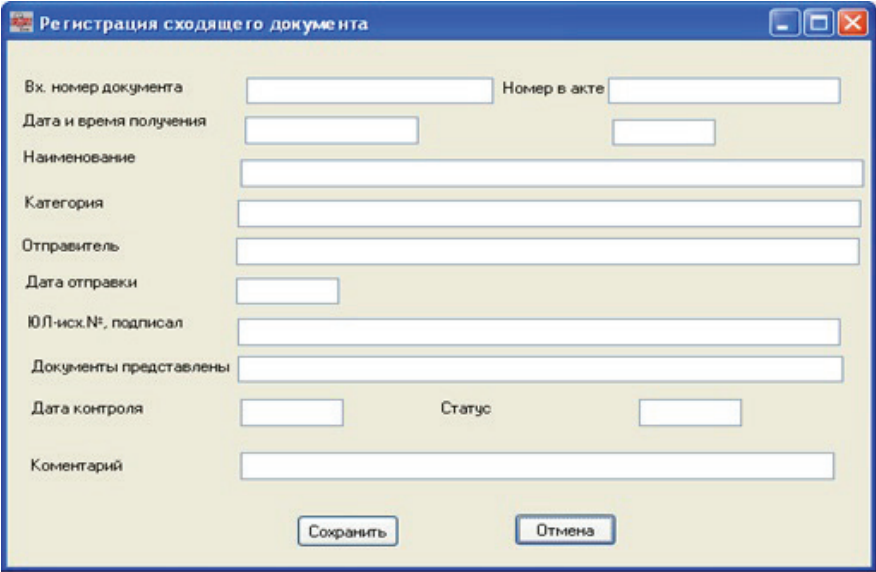


Рис. 6.6. Основные классы, описанные в BPsim.SD

Проектирование формы регистрации входящего документа заключается в размещении пользователем на шаблоне формы компонентов, необходимых для отображения всех атрибутов, связанных с данной формой классов. На рис. 6.7 приведен пример прототипа формы регистрации входящего документа, которая получена в BPsim.SD.

Спроектированные формы дорабатываются программистами. На рис. 6.8 приведен пример формы, спроектированной программистом по прототипу (рис. 6.7).



Регистрация входящего документа

Вх. номер документа Номер в акте

Дата и время получения

Наименование

Категория

Отправитель

Дата отправки

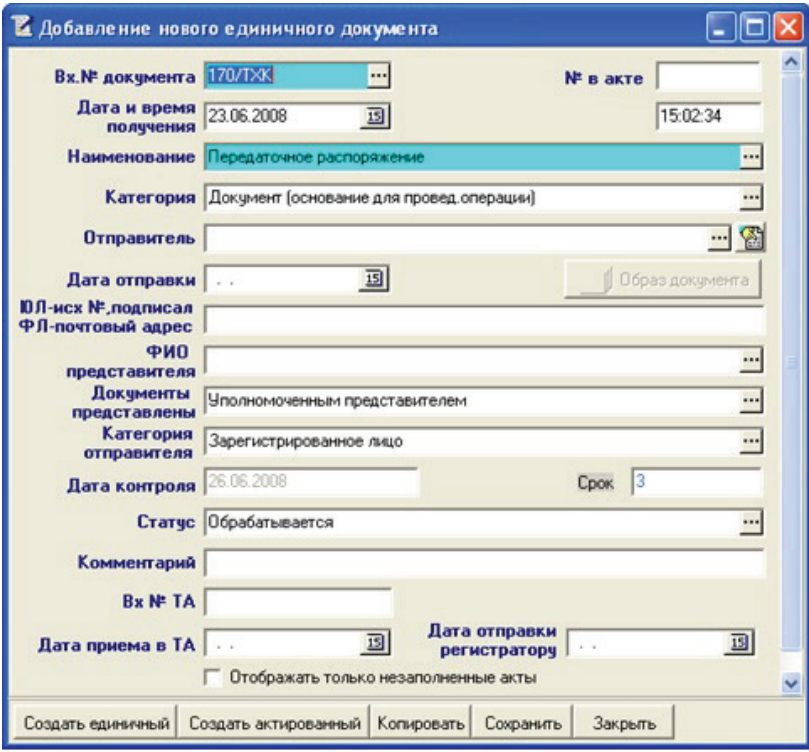
ЮЛ-иск.№, подписал

Документы представлены

Дата контроля Статус

Комментарий

Рис. 6.7. Прототип формы регистрации входящего документа



Добавление нового единичного документа

Вх.№ документа № в акте

Дата и время получения

Наименование

Категория

Отправитель

Дата отправки

ЮЛ-иск №,подписал

ФИО

представителя

Документы представлены

Категория отправителя

Дата контроля Срок

Статус

Комментарий

Вх № ТА

Дата приема в ТА Дата отправки регистратору

☐ Отображать только незаполненные акты

Рис. 6.8. Диалоговая форма регистрации входящего документа

BPsim.SD предлагает пользователю следующие возможности в рамках процесса проектирования интерфейса будущей ИС:

- описание бизнес-процессов, автоматизируемых создаваемым ПО, с помощью диаграмм стандарта DFD. Диаграммы DFD можно декомпонировать до любого уровня детализации;
- описание функций, выполняемых пользователями системы в рамках автоматизируемых процессов, с помощью диаграмм прецедентов. Имеется возможность строить диаграмму прецедентов с нуля или с помощью автоматизированного конвертирования из DFD-диаграммы, на которой выбранным из списка существующих на диаграмме DFD внешним сущностям ставятся в соответствие актеры, а функциям — прецеденты. Полученные таким образом диаграммы дальше можно редактировать. Допускается строить неограниченное количество диаграмм прецедентов для каждой диаграммы DFD;
- для каждого прецедента, выполняемого пользователем, может быть дано описание последовательности элементарных операций в системе. Для этого предназначены диаграммы последовательности, которые могут быть созданы для любой функции диаграммы прецедентов. Система позволяет частично автоматизировать этот процесс: разработчику предлагается перенести на диаграмму последовательности выбранных актеров с диаграммы прецедентов. Для каждого объекта диаграммы последовательности определяется один из четырех типов по принадлежности к определенному пакету (границы, актеры, управление, бизнес-объекты);
- создание диаграммы классов и сопоставление объектов диаграммы последовательности (кроме границ) с классами этой диаграммы;
- проектирование визуальных форм моделируемого ПО (границ диаграммы последовательности) — размещение на форме компонентов, привязка к компонентам методов и свойств классов и сохранение моделей форм в форматах.dfm и.pas. В дальнейшем данные модели могут быть переданы программисту для импорта в среду Delphi и последующей проработки алгоритмов;
- формирование отчетов о созданном проекте с изображениями спроектированных диаграмм и форм. Предусмотрен вывод отчетов на печать и экспорт в Word;
- сохранение проекта на сервере и загрузка с сервера MS SQL Server для редактирования. При сохранении проекта пользователем присваивается ему уникальное имя. По этому имени впоследствии и осуществляется загрузка проекта.

7. Описание системы технико-экономического проектирования BPsim 3

7.1. Назначение и логическая структура программы

Программа BPsim3 предназначена для решения задач технико-экономического проектирования и динамического бизнес-моделирования организационно-технических систем. Данный программный комплекс позволяет проектировать следующие элементы организационно-технической системы (ОТС) [14]:

- функциональную структуру проектируемой системы (в виде диаграмм потока данных (DFD) и прецедентов);
- концептуальную модель предметной области (в виде диаграммы классов, экземпляров классов и концептуальных графов, реализованных в виде семантической сети);
- сценарий (алгоритм) решения задачи технико-экономического проектирования и бизнес-моделирования (в виде диаграммы поиска решения, реализованной на основе диаграммы последовательности).

Программа решает задачи динамического интеллектуального моделирования мультиагентных процессов принятия решений, задачи системного анализа (обследования) и разработки моделей производственных, ОТС и бизнес-систем, анализа узких мест, реинжиниринга, оптимизации.

Основными понятиями в BPsim3 являются «предметная область» и «модель». Предметная область позволяет разделить все переменные системы на группы по предметным областям. В рамках каждой предметной области пользователь может создать несколько разных моделей.

Для того чтобы создать предметную область, надо в главном меню выбрать ПО\Новая ПО появится окно Добавить предметную область (рис. 7.1).

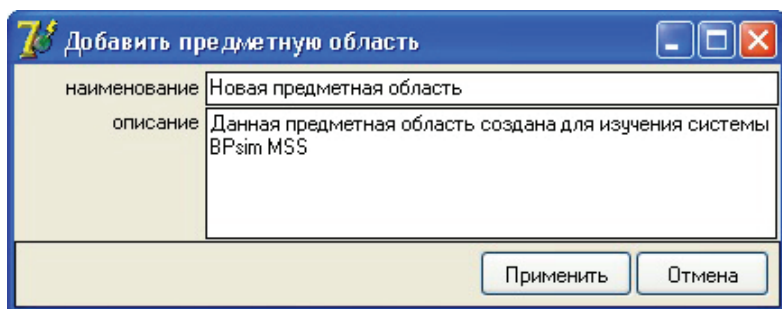


Рис. 7.1. Окно Добавить предметную область

При помощи панели инструментов следует создать новую или загрузить ранее созданный проект с сервера (рис. 7.2).

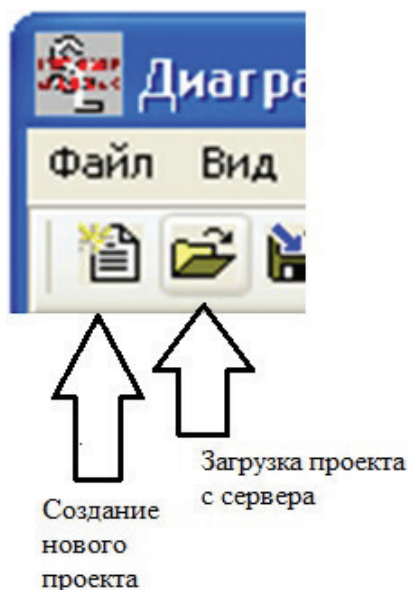


Рис. 7.2. Панель инструментов

Появится следующее диалоговое окно (рис. 7.3), в котором следует выбрать необходимое действие. С помощью двойного щелчка мыши по объектам таблицы откроются существующие на сервере проекты, а по кнопке Новый проект — проект будет создан заново (рис. 7.4).

После открытия проекта работа осуществляется в двух режимах: модель или диаграмма классов; установка текущего режима — через главное меню Вид или через панель инструментов (рис. 7.5–7.6).

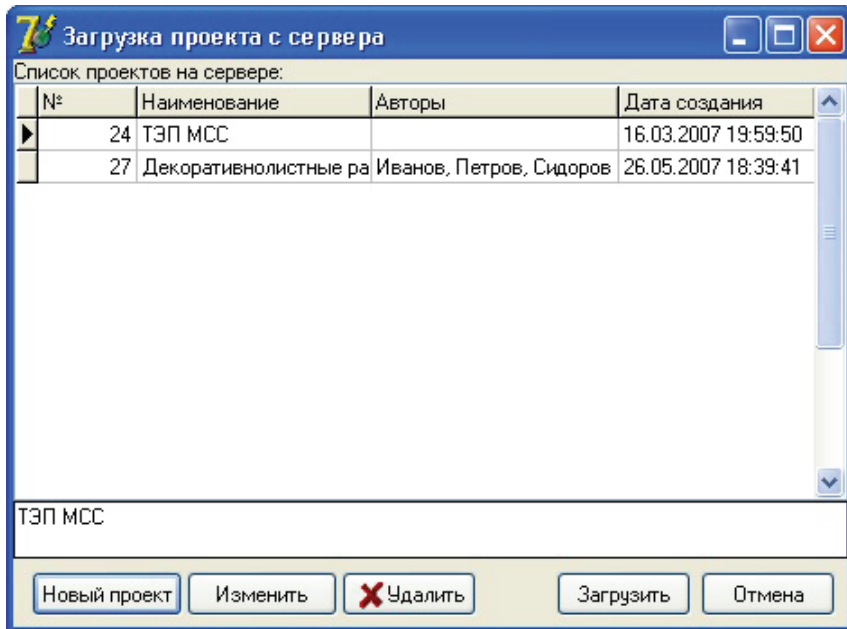


Рис. 7.3. Окно загрузки проекта

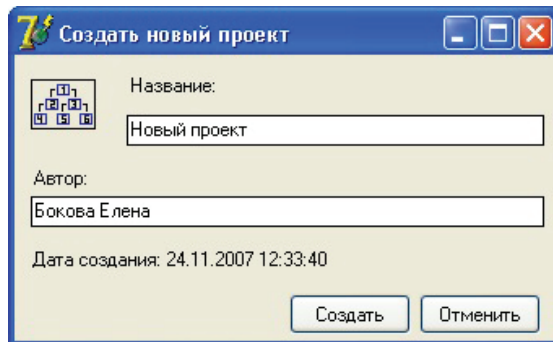


Рис. 7.4. Окно создания нового проекта

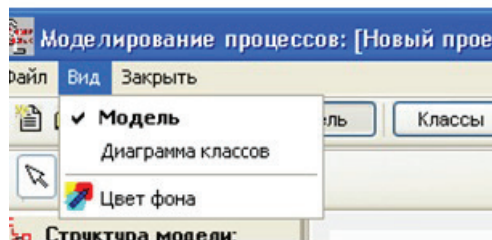


Рис. 7.5. Меню Вид

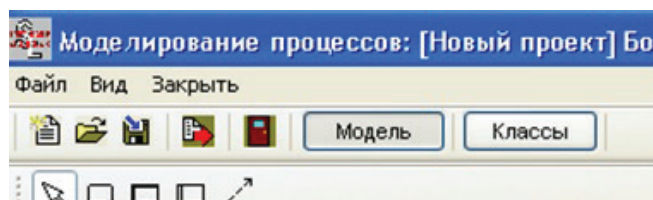


Рис. 7.6. Панель инструментов

7.2. Диаграммы и программы BPsim

7.2.1. Диаграммы DFD

С точки зрения системного анализа функциональную структуру системы следует описывать до проектирования концептуальной модели предметной области.

Возможны варианты, когда после создания классов надо перейти к построению диаграммы DFD.

Особенности взаимодействия элементов моделируемой системы могут быть представлены на диаграмме DFD. Временной аспект поведения может иметь существенное значение при моделировании синхронных процессов, описывающих взаимодействия объектов. Строится она в режиме Модель (при выборе в пункте главного меню Вид).

Стандарт DFD представляет модельную систему как сеть связанных между собой работ. Важной особенностью этого стандарта является то, что последовательность передачи управления от блока к блоку не определена. Все эти объекты можно создать при помощи меню панели инструментов (рис. 7.7).

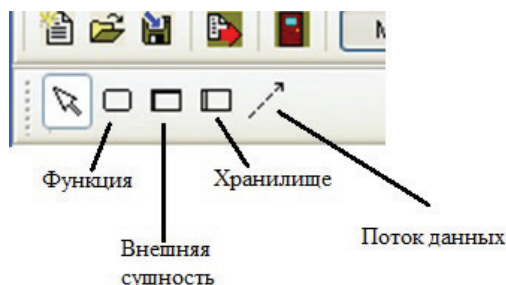


Рис. 7.7. Меню панели инструментов создания объектов

На рис. 7.8 изображена диаграмма DFD, выполненная в программе BPsim3.

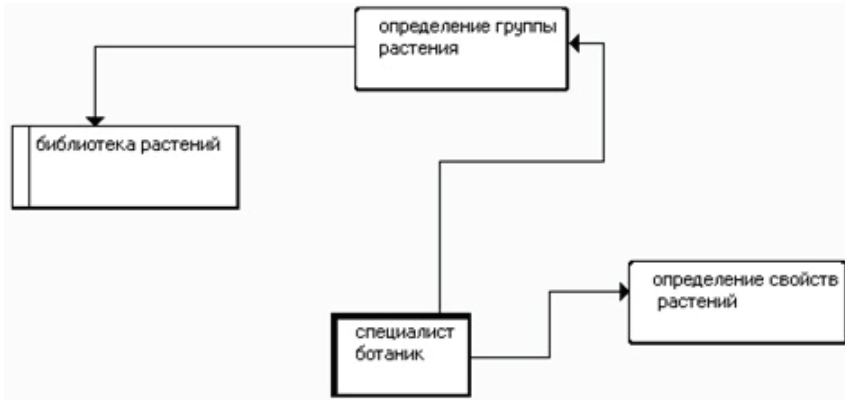


Рис. 7.8. Диаграмма стандарта DFD

При создании функций существует возможность декомпозиции (рис. 7.9).

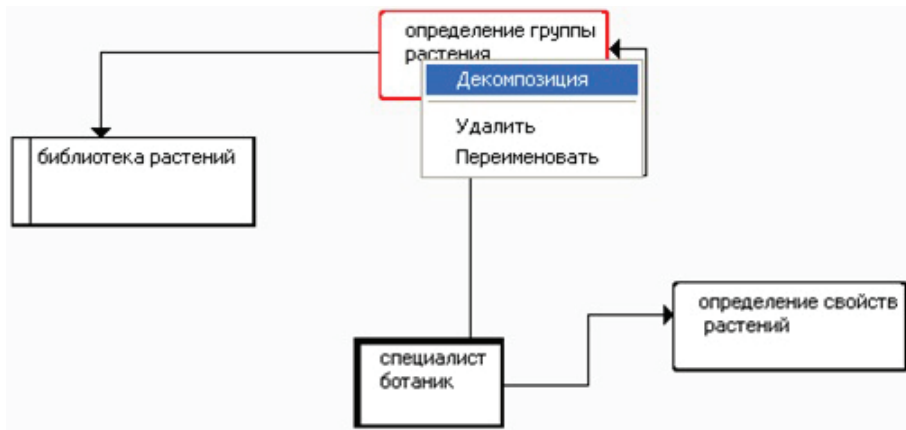


Рис. 7.9. Декомпозиция выполняемых функций (процессов)

Системой предусмотрено задание количества элементов декомпозиции (рис. 7.10).

При создании хранилищ предусмотрен выбор из ранее объявленных объектов (рис. 7.11–7.12).

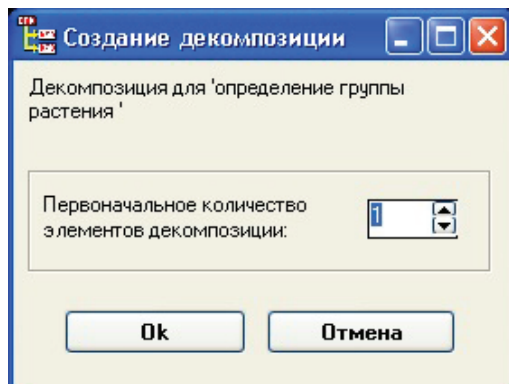


Рис. 7.10. Определение количества элементов декомпозиции

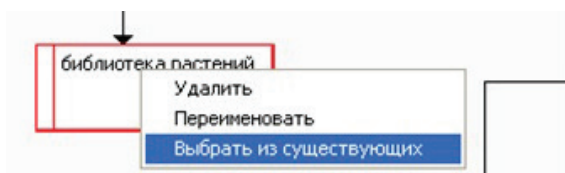


Рис. 7.11. Контекстное меню Хранилища

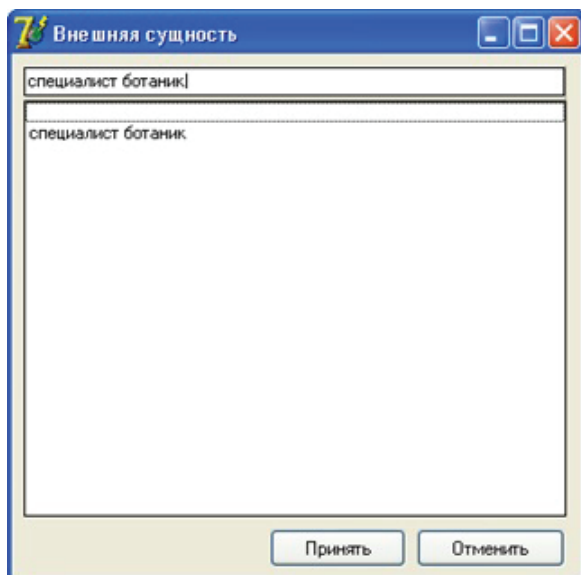


Рис. 7.12. Выбор объектов из хранилища данных

При создании внешних сущностей предусмотрен выбор из ранее объявленных объектов (рис. 7.13–7.14).

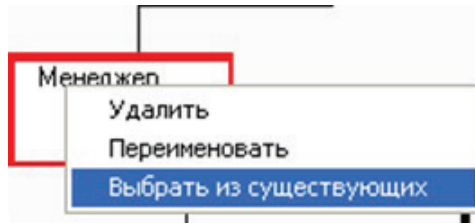


Рис. 7.13. Контекстное меню Внешняя сущность

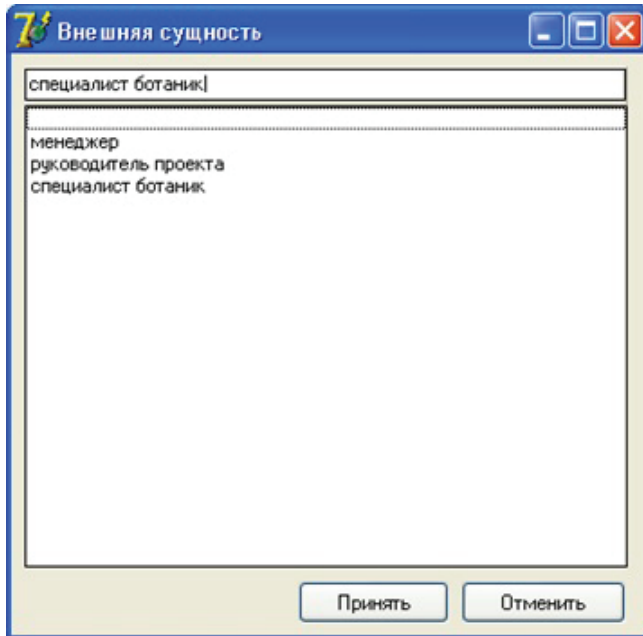


Рис. 7.14. Внешняя сущность. Пример выбора объектов

При создании потока данных предусмотрено задание, такое как на рис. 7.15. Тип потока данных демонстрирует функциональную особенность связи объектов на диаграмме по аналогии с ключевыми запросами языка Transact SQL. Определение типа потока не налагает никакой функциональной зависимости на объекты BPsim.MSS, это свойство введено для наилучшего восприятия графического представления модели и описания структуры создаваемой экспертной системы технико-экономического проектирования (ТЭП). Существуют следующие типы потоков данных:

- Insert (добавление);
- Delete (удаление);

- Update (обновление);
- Select (выборка);
- Общий.

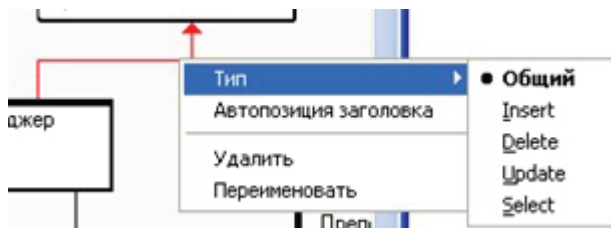


Рис. 7.15. Пример задания типа потока данных

7.2.2. Диаграммы прецедентов

После создания диаграмм DFD (функциональной структуры) их необходимо автоматически конвертировать в диаграммы прецедентов языка UML. Этот процесс осуществляется путем преобразования диаграмм в диаграммы прецедентов, описывающие функциональное назначение системы (т. е. то, что система будет делать в процессе функционирования). Каждая диаграмма прецедентов позволяет создать соответствующую диаграмму последовательности. Преобразование происходит по нажатию кнопки Конвертировать (рис. 7.16).

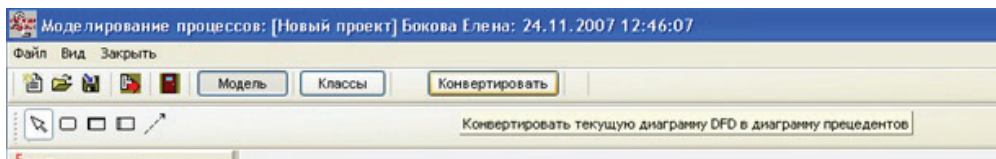


Рис. 7.16. Конвертирование диаграммы

Далее появляется диалоговое окно, в котором следует выбрать, какие именно объекты должны быть отображены или сохранены в диаграмме после конвертирования (рис. 7.17).

Результат конвертирования представлен на рис. 7.18. После преобразований исходная DFD диаграмма сохраняется в проекте и доступна к использованию и изменению.

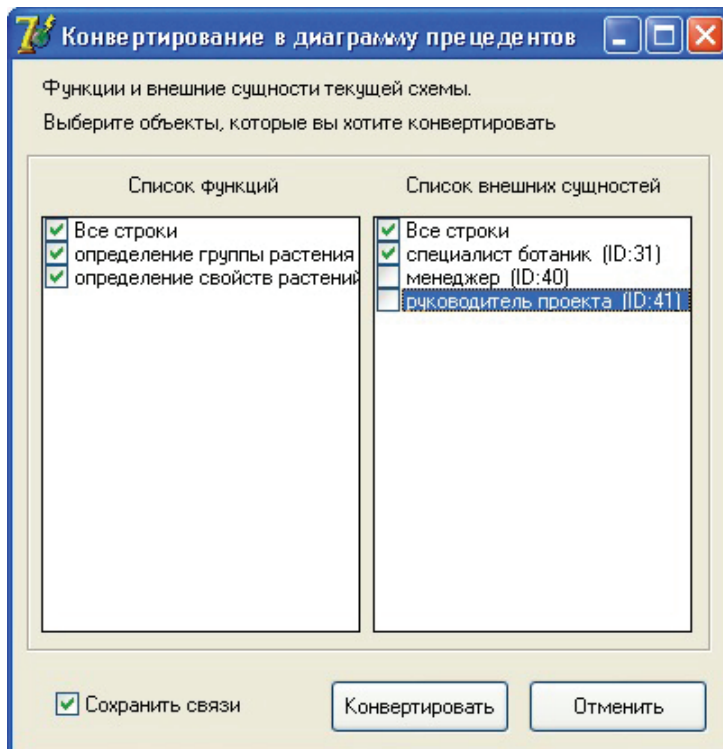


Рис. 7.17. Пример выбора объектов для преобразования

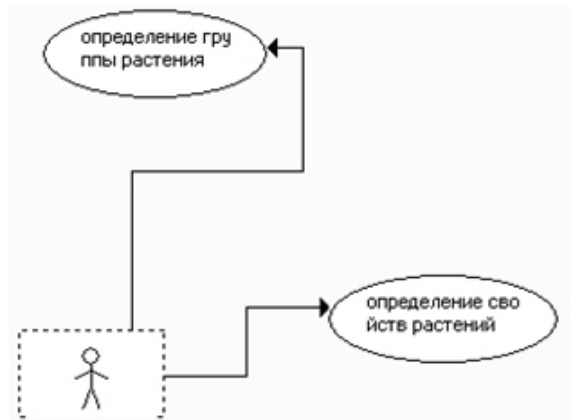


Рис. 7.18. Диаграмма прецедентов

Полученную в результате конвертирования диаграмму прецедентов можно изменять. Диаграмма прецедентов представляет собой одну из пяти типов диаграмм, применяемых в UML для моделирования ди-

намических аспектов системы (остальные четыре типа — это диаграммы деятельности, состояний, последовательностей и кооперации). Диаграммы прецедентов играют основную роль в моделировании поведения системы, подсистемы или класса. Каждая такая диаграмма показывает множество прецедентов, актеров и отношения между ними.

Диаграммы прецедентов применяются для моделирования вида системы с точки зрения прецедентов (или вариантов использования). Чаще всего это предполагает моделирование контекста системы, подсистемы или класса либо моделирование требований, предъявляемых к поведению указанных элементов.

Диаграммы прецедентов имеют большое значение для визуализации, специфицирования и документирования поведения элемента. Они облегчают понимание систем, подсистем или классов, представляя взгляд извне на то, как данные элементы могут быть использованы в соответствующем контексте. Кроме того, такие диаграммы важны для тестирования исполняемых систем в процессе прямого проектирования и для понимания их внутреннего устройства при обратном проектировании.

В системе BPsim3 диаграммы прецедентов определяют перечень процессов принятия решений, для автоматизированной поддержки которых используется аппарат диалоговых экспертных систем. Рассмотрим элементы этой диаграммы.

Актер — любая сущность, взаимодействующая с системой извне, или множество логически связанных ролей, исполняемых при взаимодействии с прецедентами (функциями) (рис. 7.19). Его стандартным графическим обозначением на диаграммах является фигурка человечка, однако актером может быть не только человек, но и техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определяет сам разработчик.

Функция (аналог прецедента) — прецеденты (use case) — это описание множества последовательностей действий (включая их варианты), которые выполняются системой для того, чтобы актер получил результат, имеющий для него определенное значение. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой, это одна из важнейших особенностей разработки прецедентов. Стандартным графическим обозначением прецедента на диаграммах является эллипс, внутри которого содержится крат-

кое название прецедента или имя в форме глагола с пояснительными словами (рис. 7.20).

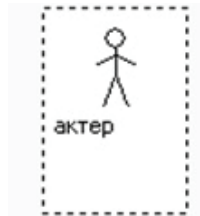


Рис. 7.19. Схематическое изображение актера на диаграмме

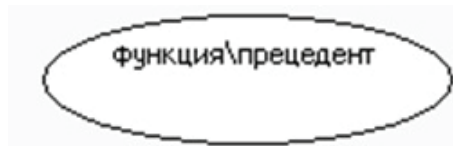


Рис. 7.20. Схематическое изображение прецедента на диаграмме

Поток данных описывает взаимодействие актеров и прецедентов (рис. 7.21).

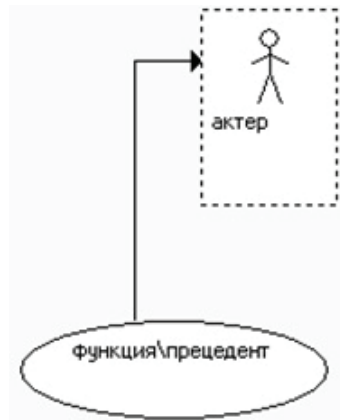


Рис. 7.21. Схематическое изображение связи на диаграмме

Добавление объектов на диаграмму происходит при помощи меню панели инструментов (рис. 7.22).

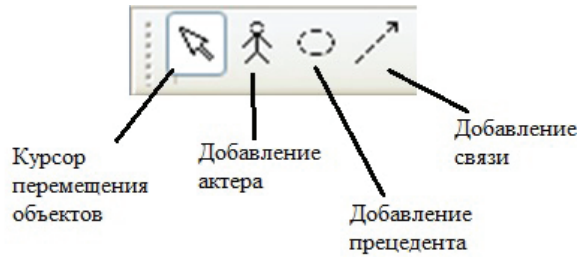


Рис. 7.22. Пример добавления объектов на диаграмму прецедентов

7.2.3. Подсистема диаграммы классов

В данной подсистеме возможно построение различных диаграмм-классов, прецедентов, последовательности. Одной из важных целей данной подсистемы является поиск решений с использованием диаграмм последовательности. Именно эта возможность будет подробно рассмотрена далее.

Для выполнения поиска решения необходимо последовательно выполнить следующие шаги.

1. Построение диаграммы классов состоит из следующих действий:

- создание классов;
- описания свойств класса (атрибутов);
- задания методов класса;
- описания экземпляров класса.

2. Построение семантической сети.

3. Построение диаграммы последовательности.

4. Проигрывание (поиск решения) диаграммы последовательности.

Рассмотрим эти шаги.

Создание нового класса осуществляется с помощью кнопок нижнего меню панели инструментов (рис. 7.23).

Для каждого класса доступно контекстное меню (рис. 7.24).

Рассмотрим свойства класса.

Общие свойства класса (рис. 7.25):

- Имя класса (задается пользователем);
- Идентификатор класса (задается программой);
- Системное обозначение (задается пользователем). Одним из наиболее важных является заполнение этого поля (в дальнейшем каждому классу будет поставлена в соответствие таблица).

! Необходимо задавать латинскими буквами:

- Тип
 - ✓ класс,
 - ✓ класс-наполнитель;
- Описание (поясняющая информация);
- Использовать в диаграмме Ганта.

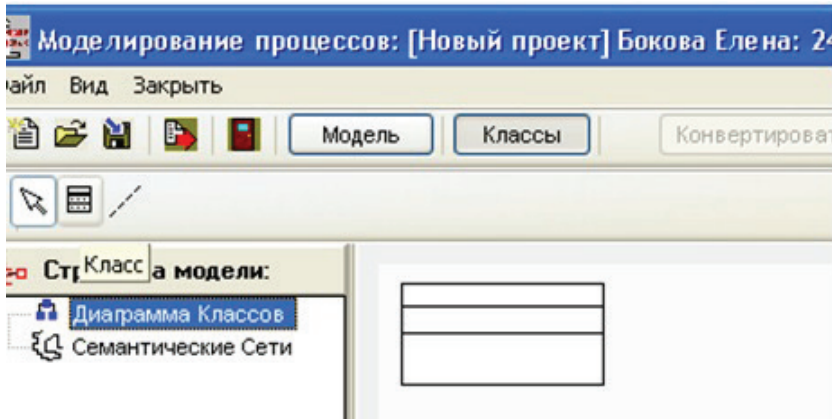


Рис. 7.23. Создание нового класса

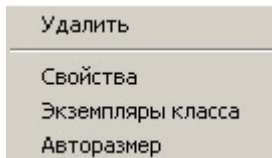


Рис. 7.24. Контекстное меню класса

Диаграмма Ганта (англ. *Gantt chart*, также ленточная диаграмма) — это популярный тип столбчатых диаграмм, который используется для иллюстрации плана, графика работ по какому-либо проекту. Является одним из методов планирования проектов.

Диаграмма Ганта представляет собой отрезки (графические плашки), размещенные на горизонтальной шкале времени. Каждый отрезок соответствует отдельной задаче или подзадаче. Задачи и подзадачи, составляющие план, размещаются по вертикали. Начало, конец и длина отрезка на шкале времени соответствуют началу, концу и длительности задачи. Диаграмма может использоваться для представления текущего состояния выполнения работ: часть прямоугольника,

отвечающего задаче, заштриховывается, отмечая процент выполнения задачи; показывается вертикальная линия, отвечающая моменту «сегодня» (рис. 7.26).

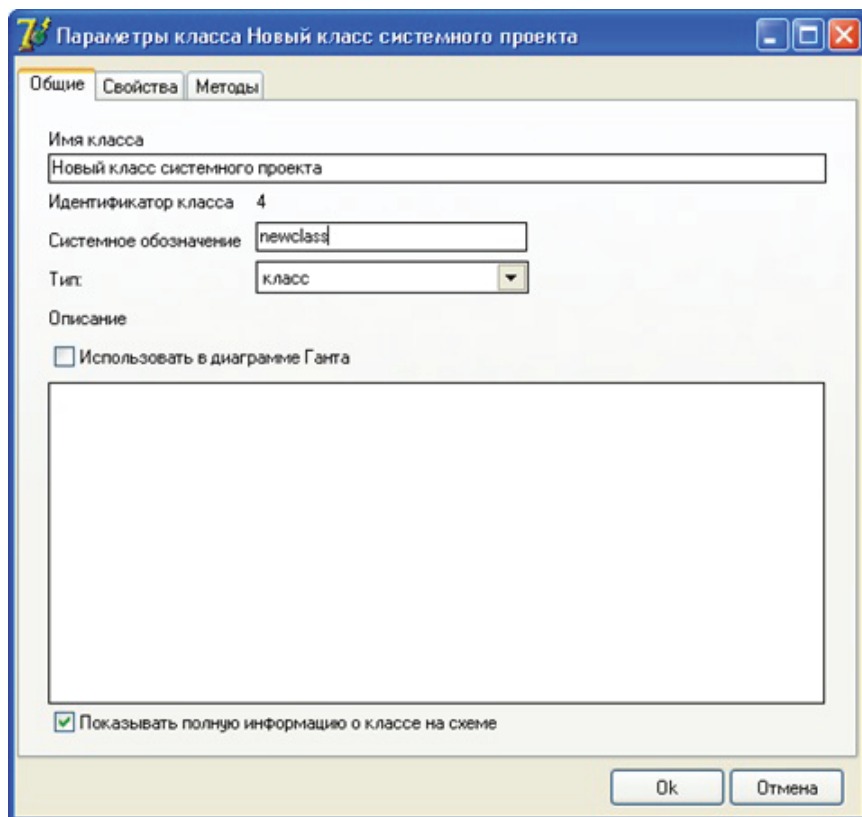


Рис. 7.25. Общие свойства класса

Настройка Показывать полную информацию о классе на схеме позволяет определять полноту информации. При ее установке на схеме видны не только имя класса и созданные параметры, но и тип созданных параметров (рис. 7.27–7.28).

На вкладке Свойства каждому классу ставится в соответствие таблица (рис. 7.29), каждая строка которой является отдельным свойством. Рассмотрим колонки таблицы, которые представляют собой характеристики свойства.

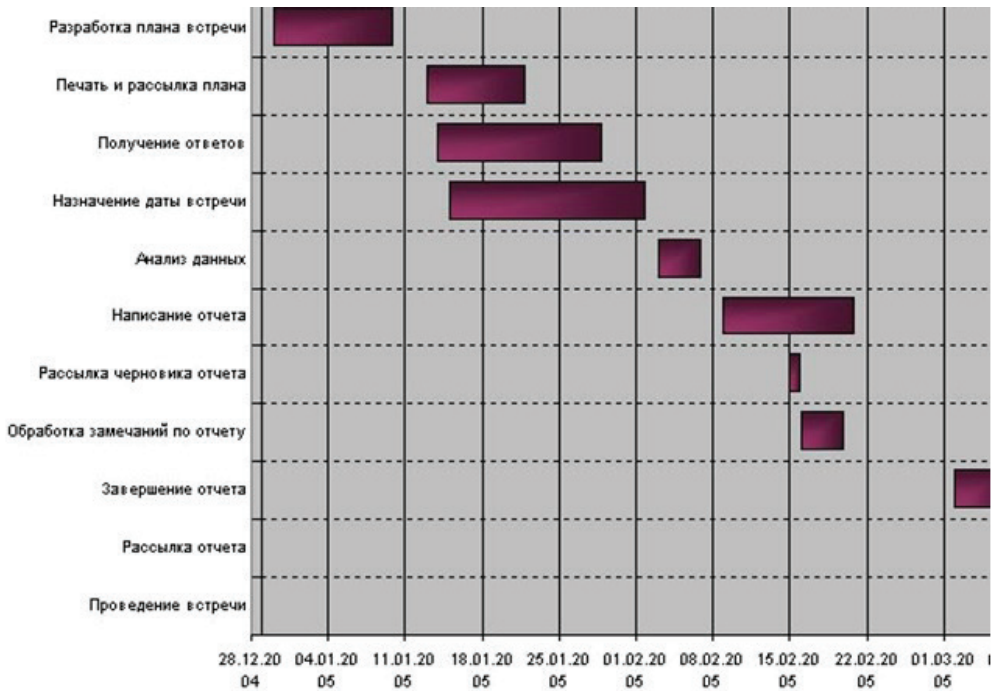


Рис. 7.26. Пример диаграммы Ганта

(Да/нет)
■ значение : Логическое

Рис. 7.27. Отображение полной информации о классе

(Да/нет)
■ значение

Рис. 7.28. Отображение неполной информации о классе

Важным для дальнейшей работы является заполнение поля Обозначение (свойства в дальнейшем будут использованы в качестве полей таблицы, реализующей класс).

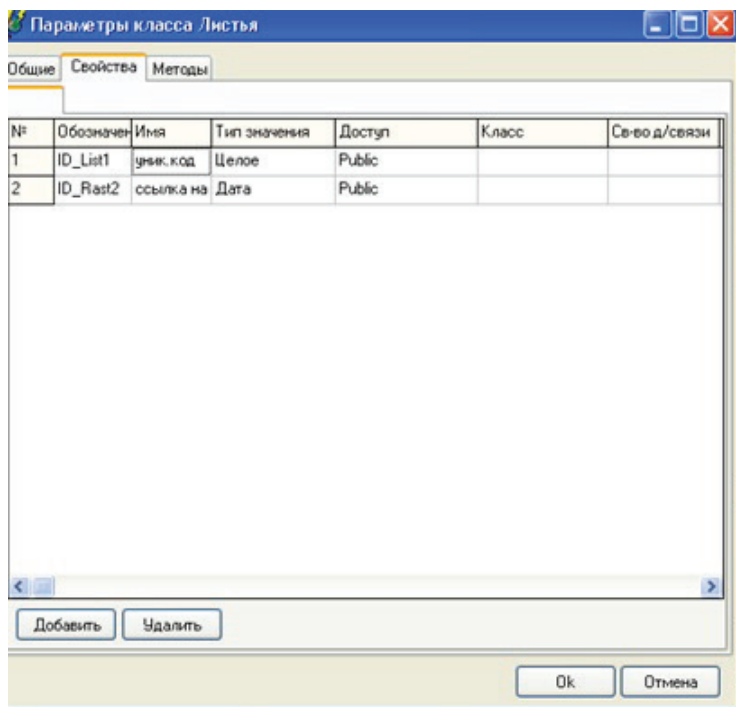


Рис. 7.29. Пример задания параметров класса

! Необходимо задавать латинскими буквами:

- Имя (уникальное имя параметра);
- Тип значения (выбирается тип вводимого параметра из предложенных системой) (рис. 7.30);

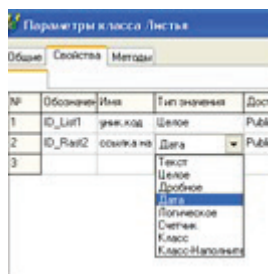


Рис. 7.30. Тип значения

- Доступ (ограничивает доступ к данному параметру) (рис. 7.31);

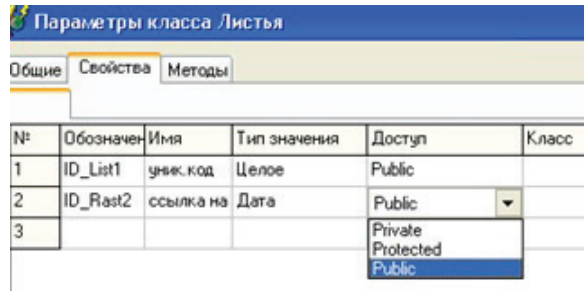


Рис. 7.31. Доступ

- Класс — в программе между классами возможна организация связи. Данная характеристика задает, с каким именно классом происходит это соединение. Если провести аналогию с организацией структуры таблиц MS SQL Server, то данная характеристика может быть интерпретирована как задание внешнего ключа таблицы;
- Свойство для связи задает, по какому именно полю происходит связь двух таблиц классов. Графически связь между таблицами может быть представлена при помощи добавления соответствующих связующих линий, вызываемых в меню панели инструментов (рис. 7.32).

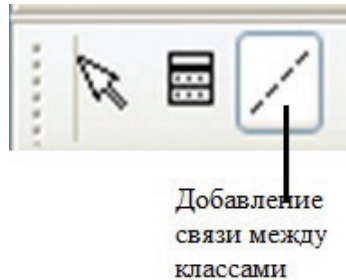


Рис. 7.32. Добавление связи между классами

Например, пусть будут созданы два класса: «Факультет» и «Дисциплины». Нам необходимо связать их по полю «Дисциплина». Для этого зададим следующие параметры связи (рис. 7.33–7.34).

№	Обозначение	Имя	Тип значения	Доступ	Класс	Связь д/связи
1	ID_facult	Факультет	Текст	Public		
2	ID_spe	Специальность	Текст	Public		
3	ID_disc	Дисциплина	Текст	Public	Дисциплины	Дисциплины

Рис. 7.33. Пример задания параметров класса «Факультет»

№	Обозначение	Имя	Тип значения	Доступ	Класс	Связь д/связи
1	ID_disc	Дисциплина	Текст	Public		
2	ID_pre	Преподаватель	Текст	Public		

Преподаватель

Рис. 7.34. Пример задания параметров класса «Дисциплина»

На вкладке Методы осуществляется построение методов класса. Метод представляет собой запрос, написанный на языке T-SQL, выполняющий определенные преобразования над таблицей класса. Двойной клик на колонке SQL-текст вызывает окно, в котором осуществляется написание запроса метода и внесение информации о параметрах запроса. При написании методов следует иметь в виду особенности, указанные ниже.

Информация о диаграмме классов хранится в виде некоторой базы данных, в которой каждому классу (при непустом системном обозначении) ставится в соответствие временная таблица. Например, при системном обозначении класса *Rastenia* наименование временной таблицы соответственно будет *#Rastenia*. Наименование полей этой временной таблицы, а также типы полей совпадают с наименованиями свойств класса, описанных на вкладке «Свойства». Кроме того, в каждой таблице создается поле — флаг с наименованием F_Use, по умолчанию принимающий значение 0. Возможно использовать это поле для выделения экземпляров классов по различным признакам, в частности, 1 — экземпляр удовлетворяет поиску, 0 — не удовлетворяет. Возможны и другие варианты, например удаление экземпляров, не удовлетворяющих условиям поиска решения. Более подробно это будет рассмотрено далее.

При написании метода возникает необходимость в описании параметров, имена параметров описываются в нижней части формы и начинаются со знака двоеточия. Для параметров описываются наименование (что облегчит введение значений при поиске решения в дальнейшем) и значения по умолчанию, что также ускорит процесс поиска решения (рис. 7.35).



Рис. 7.35. Задание метода

На рис. 7.35 представлена таблица, описывающая методы класса. Каждая строка таблицы представляет собой один метод, колонки таблицы — характеристики метода. Рассмотрим их кратко.

- Имя задает имя выполняемым над таблицей преобразованиям;
- Тип возвращаемого значения определяет тип возвращаемого значения после выполнения метода;
- Доступ определяет доступ к просмотру данного поля;
- Условия запуска — задание условия, по которому произойдет выполнение преобразований. Поле может быть пустым, тогда операция будет выполнена в любом случае;
- SQL-текст — текст запроса на языке Transact SQL, диалоговое окно ввода формулы вызывается двойным кликом мыши (рис. 7.36).

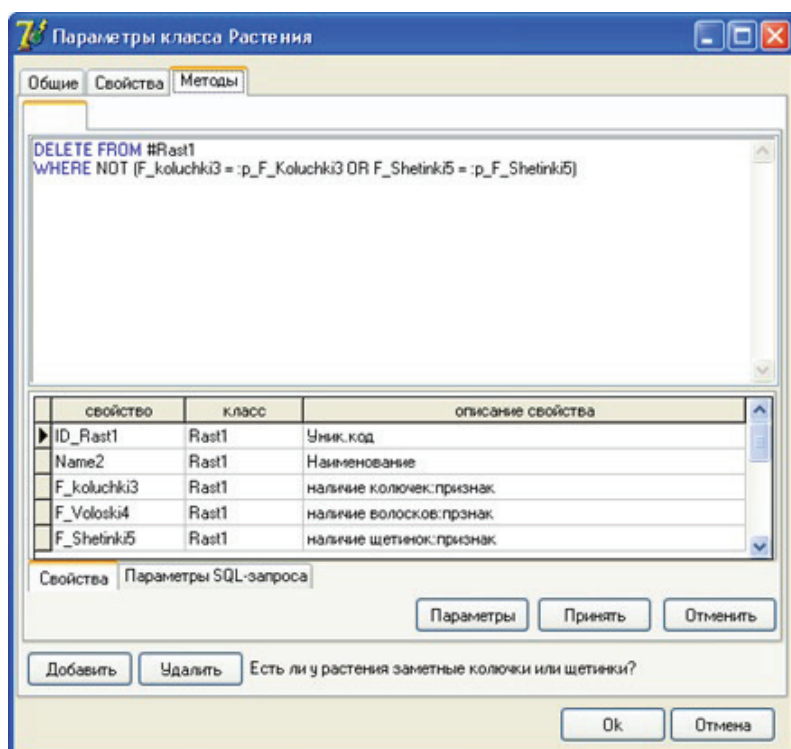


Рис. 7.36. SQL-запрос

Вкладка Свойства выводит все параметры ранее заданных классов, над которыми возможно произвести преобразования. На вкладке Параметры SQL-запроса должны быть объявлены все параметры, задей-

ствованные в запросе. Объявление происходит при помощи кнопки Добавить (рис. 7.37).

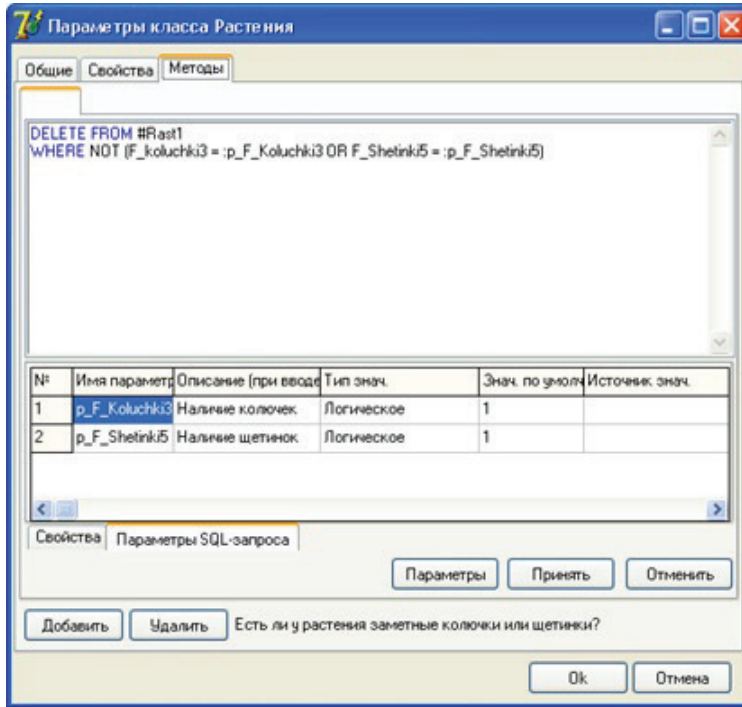


Рис. 7.37. Объявление параметров SQL-запроса

Создание экземпляров классов происходит при вызове контекстного меню щелчком по правой клавише мыши (рис. 7.38–7.39).

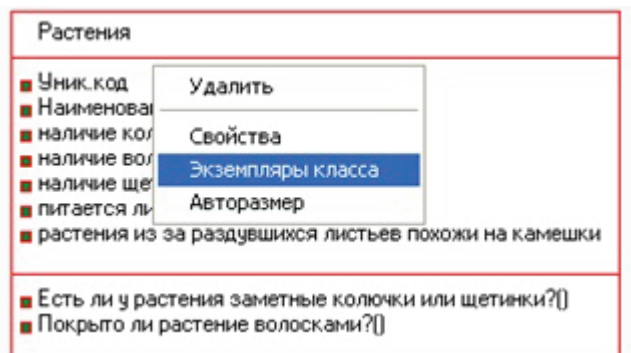


Рис. 7.38. Вызов контекстного меню

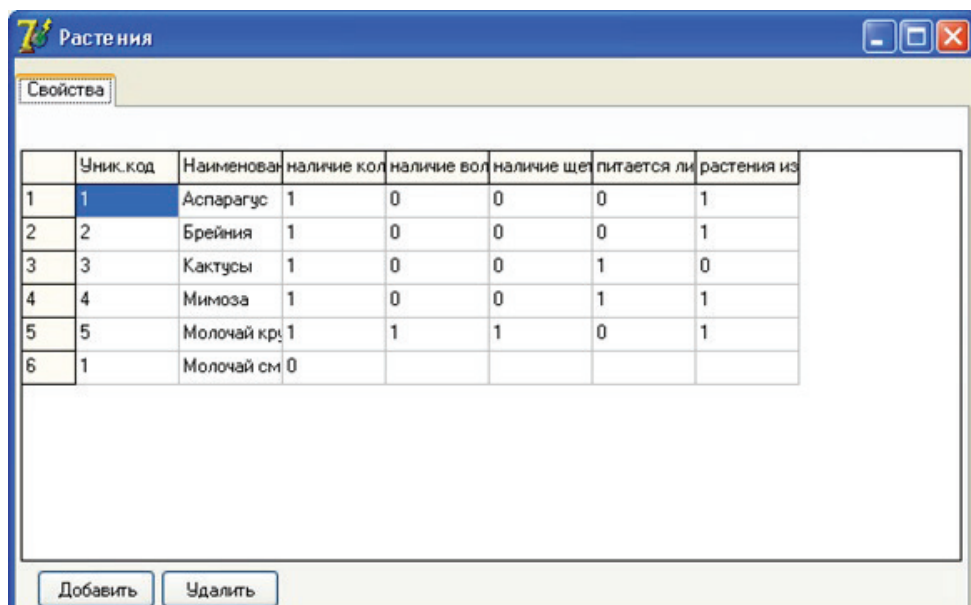


Рис. 7.39. Пример добавления нового экземпляра класса

7.2.4. Диаграмма последовательности (поиска решения)

Временной аспект поведения объектов может иметь существенное значение при моделировании синхронных процессов, описывающих взаимодействие объектов, и поэтому для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности.

Каждая диаграмма прецедентов позволяет создать соответствующую диаграмму последовательности. При построении модели осуществляется упорядочивание методов, описанных ранее в диаграмме классов. Каждый объект диаграммы последовательности должен быть поставлен в соответствие либо классу, либо методу. Также следует иметь в виду, что с использованием объектов одной диаграммы классов можно построить несколько диаграмм последовательности (рис. 7.40).

Перед созданием диаграммы последовательности программа выдает предупреждение, в котором объявлены объекты, принимающие участие в создаваемой диаграмме последовательности (рис. 7.41), пользователь может либо согласиться с предложенным списком, либо отменить некоторые объекты.

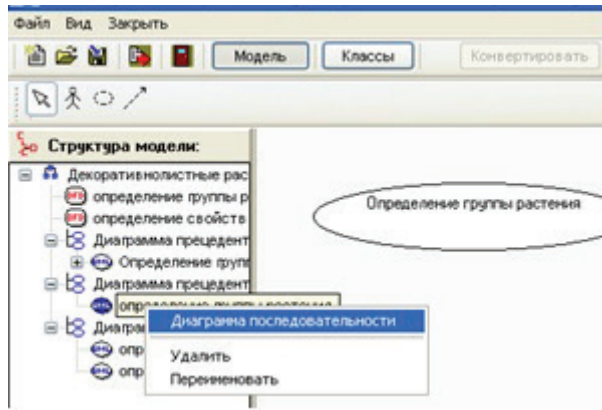


Рис. 7.40. Пример создания диаграммы последовательности

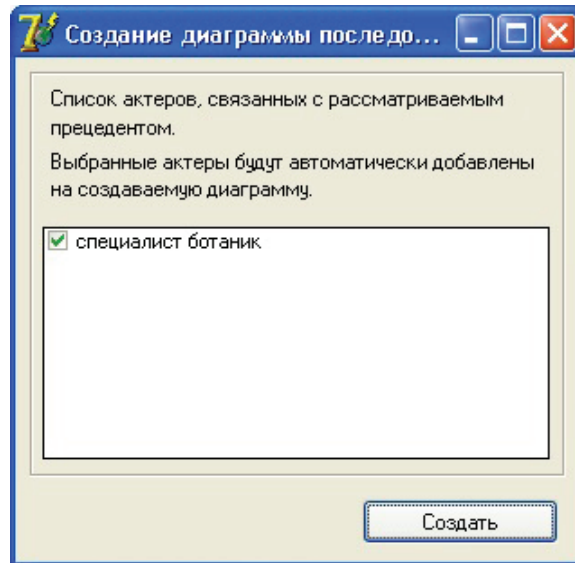


Рис. 7.41. Задание объектов диаграммы

На диаграмме последовательности изображаются исключительно те объекты, которые непосредственно участвуют во взаимодействии, и не показываются возможные статические ассоциации с другими объектами. Для диаграммы последовательности ключевым моментом является именно динамика взаимодействия объектов во времени. При этом диаграмма последовательности имеет как бы два измерения. Первое — слева направо в виде вертикальных линий, каждая из них изображает линию жизни отдельного объекта, участвующего во взаимо-

действии. Графически каждый объект изображается прямоугольником и располагается в верхней части своей линии жизни (рис. 7.42). Внутри прямоугольника записываются имя объекта и имя класса, разделенные двоеточием.

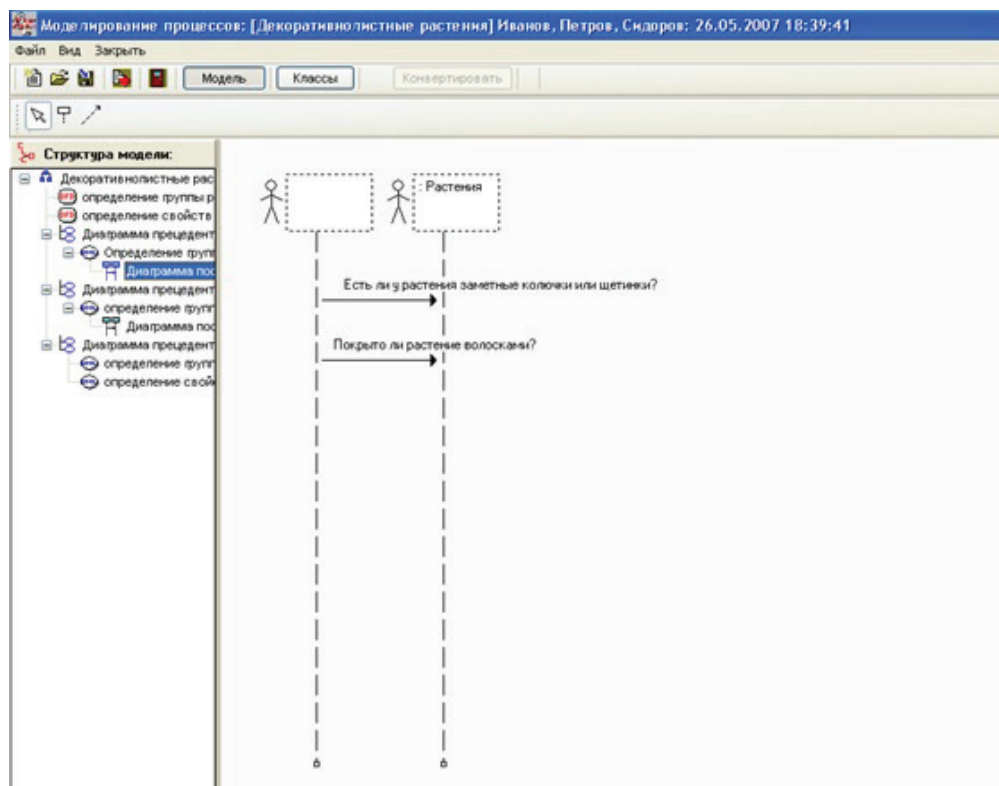


Рис. 7.42. Пример диаграммы последовательности

Крайним слева на диаграмме изображается объект, который является инициатором взаимодействия. Правее изображается другой объект, который непосредственно взаимодействует с первым. Таким образом все объекты на диаграмме последовательности образуют некоторый порядок, определяемый степенью активности этих объектов при взаимодействии друг с другом.

Второе измерение диаграммы последовательности — вертикальная временная ось, направленная сверху вниз. Начальному моменту времени соответствует самая верхняя часть диаграммы. При этом взаимодействие объектов реализуется посредством сообщений, которые по-

сылаются одними объектами другим. Сообщения изображаются в виде горизонтальных стрелок с именем сообщения и также образуют порядок по времени своего возникновения. Другими словами, сообщения, расположенные на диаграмме последовательности выше, инициируются раньше тех, которые расположены ниже. При этом масштаб на оси времени не указывается, поскольку диаграмма последовательности моделирует лишь временную упорядоченность взаимодействий типа «раньше — позже».

При создании диаграммы последовательности объекты могут быть следующих типов:

- Актер;
- Граница;
- Бизнес-объект;
- Управление;
- Неопределенный.

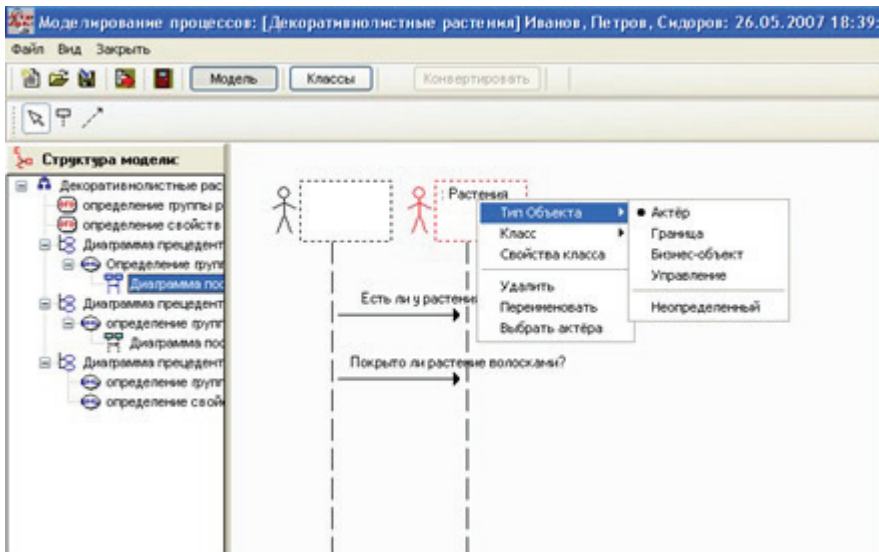


Рис. 7.43. Выбор типов объектов

Класс объекта (рис. 7.44) может быть выбран из уже существующих или создан заново.

При выборе из ранее созданных классов появляется диалоговое окно с перечнем наименований (рис. 7.45).

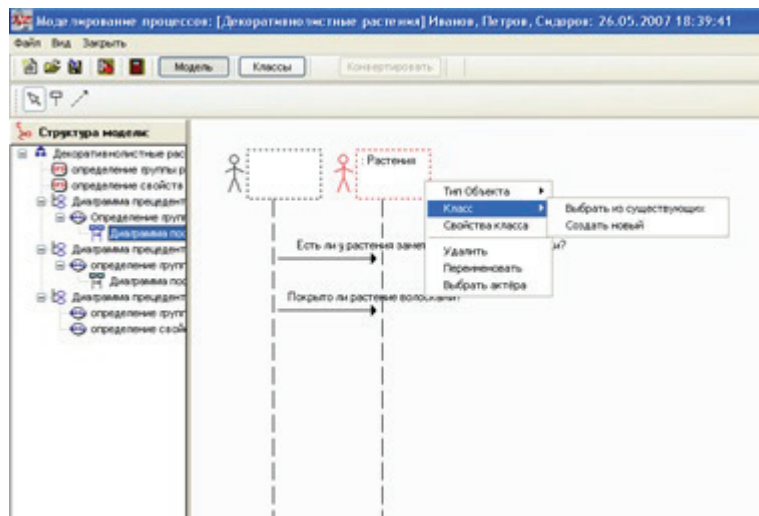


Рис. 7.44. Задание класса объекта

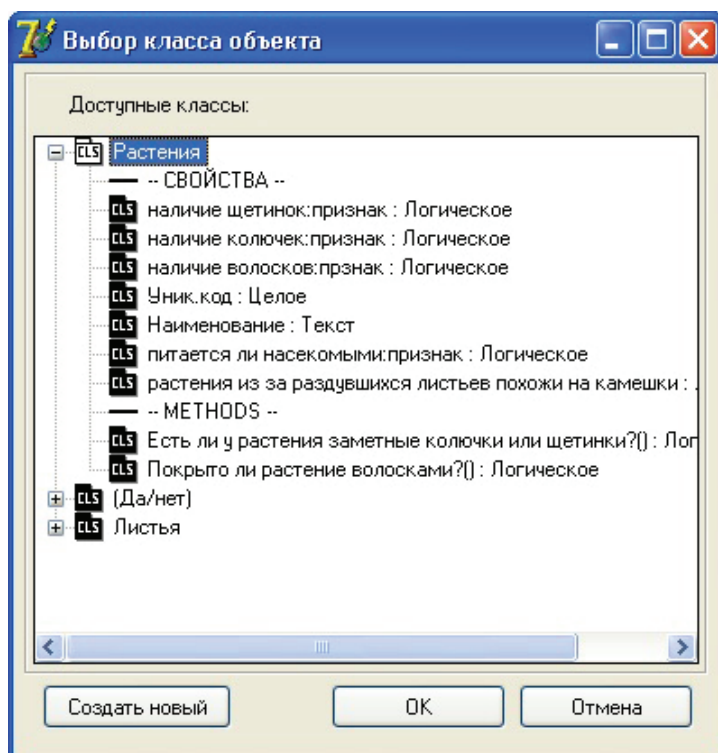


Рис. 7.45. Пример выбора класса объекта

При необходимости добавить объект нового класса появляется окно диаграммы классов (рис. 7.46).

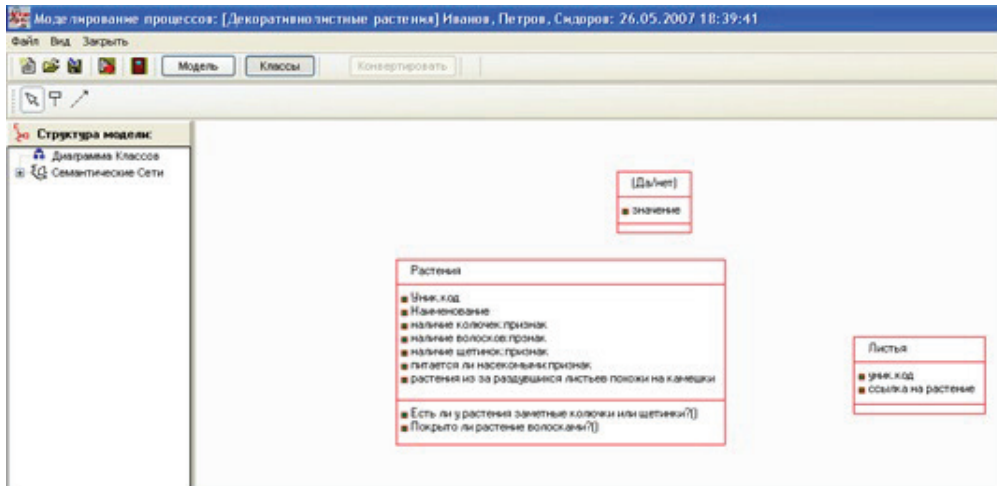


Рис. 7.46. Пример создания нового класса

Основной задачей для диаграммы последовательности является поиск решения (рис. 7.47).

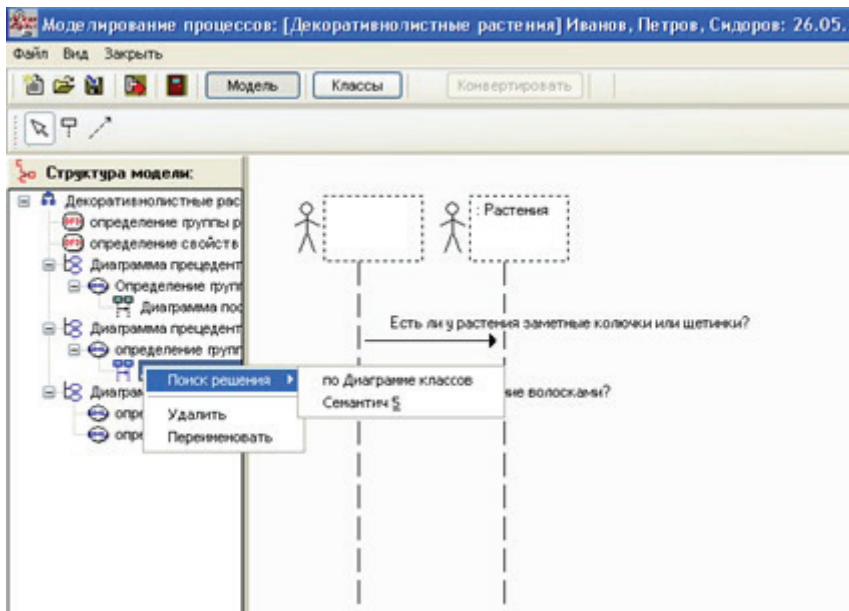


Рис. 7.47. Поиск решения

При поиске решения программа обрабатывает метод за методом (согласно диаграмме последовательности), предлагает ввести пользователю значения параметров и получить некоторое множество решений. Стоит отметить, что на форме поиска решения (рис. 7.49) можно увидеть данные всех таблиц созданной базы данных, а также задать итоговую выборку с помощью sql-запроса. Естественно, при соответствующих экземплярах класса могут получиться разные решения на выходе, они будут зависеть от входящих данных и параметров.

! Методы диаграммы поиска решения обрабатываются в порядке сверху вниз. Возможны возвраты на предыдущие методы.

Рассмотрим поиск решения на примере рассмотренной выше диаграммы последовательности. Задачи поиска основаны на методах класса, заданных ранее. Методы представлены ниже.

- Есть ли у растения заметные колючки и щетинки?
- Покрыто ли растение волосками?

Первое диалоговое окно выбора типа поиска представлено на рис. 7.48.

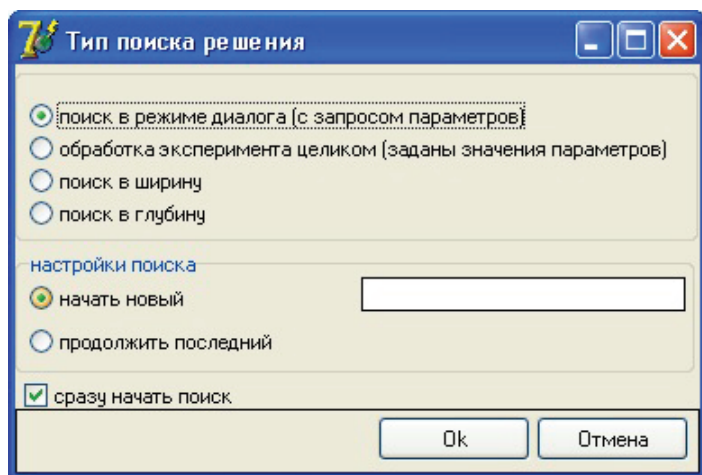


Рис. 7.48. Выбор типа поиска решения

Если нажать на кнопку ОК, то программа начнет поиск решения. Выбирается и выполняется первый метод, представленный на диаграмме последовательности. На рис. 7.49 отражен запрос параметров поиска на первом шаге.

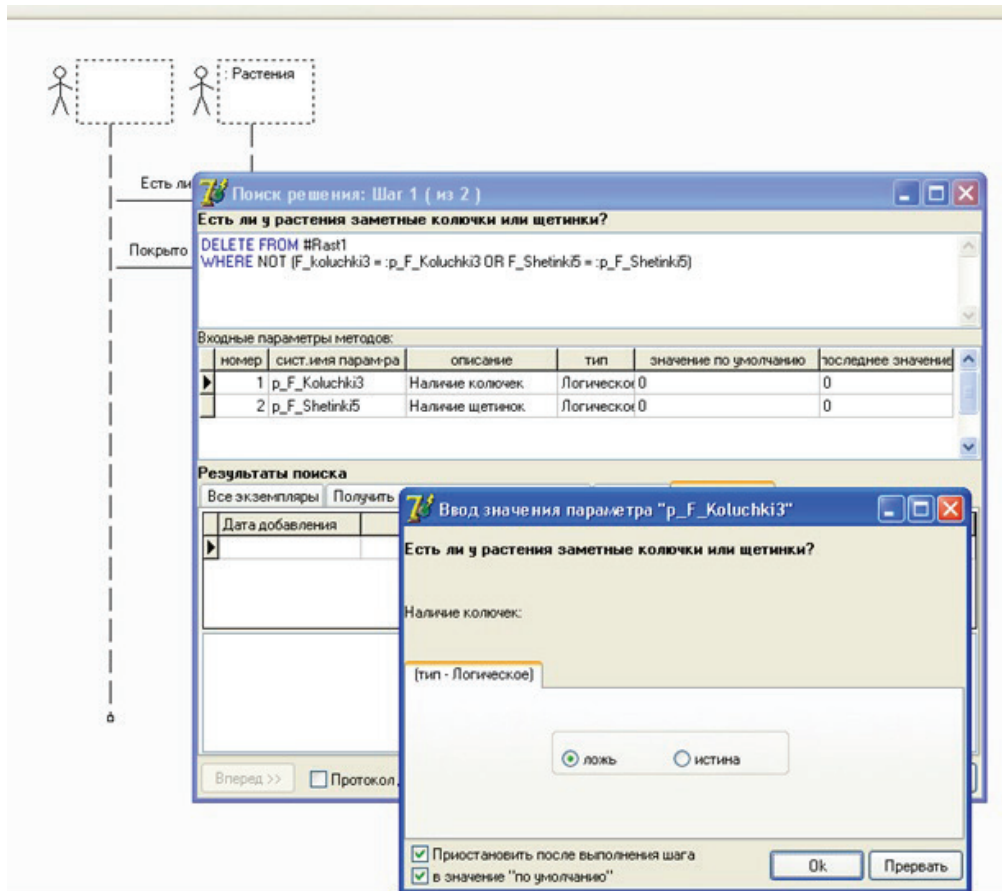


Рис. 7.49. Запрос параметров модели в диалоге с пользователем

При запросе параметров доступен к просмотру непосредственно сам текст SQL-запроса и входные параметры методов. По завершении первого шага (выполнение первого метода) можно посмотреть предварительные результаты как в виде таблицы, так и в виде текста (рис. 7.50–7.51). Для этого необходимо активировать выполнение SQL-запроса (рис. 7.52).

Результаты выполненного запроса можно сохранить в текстовом документе (рис. 7.53–7.55).

По кнопке Вперед объявляем переход ко второму методу и вводим параметры запроса (рис. 7.56).

По окончании проигрывания модели, системой выдается соответствующее сообщение (рис. 7.57).

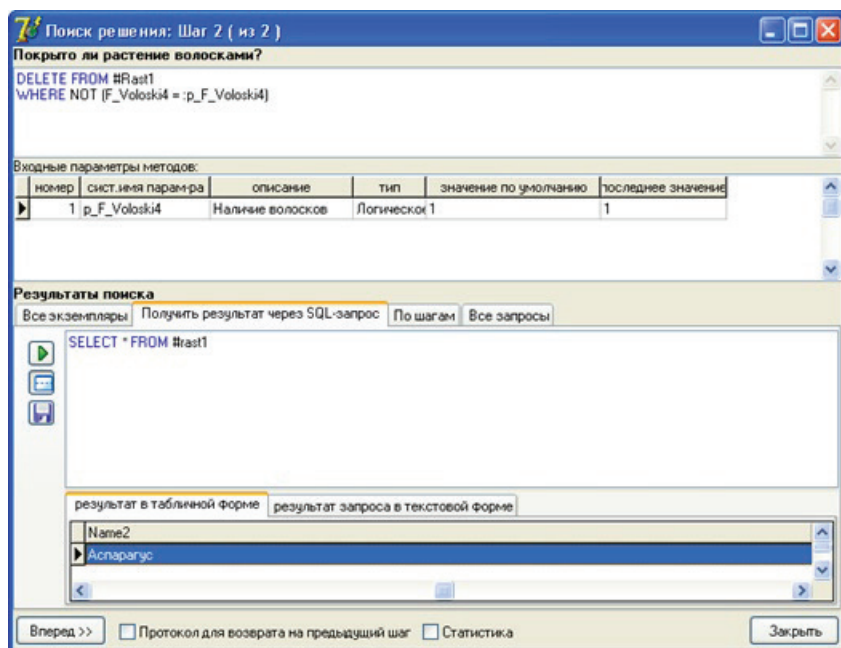


Рис. 7.50. Результаты поиска решения в форме таблицы

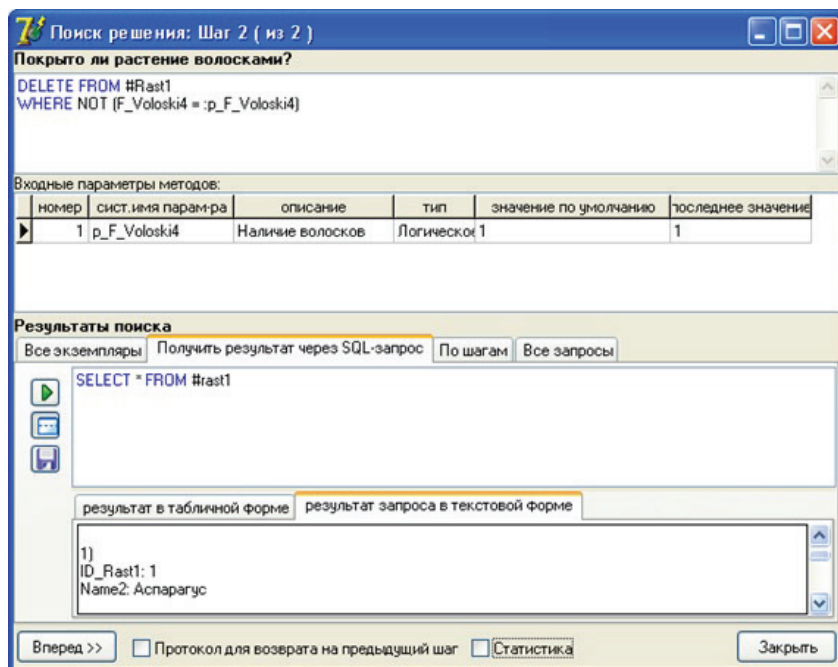


Рис. 7.51. Результаты поиска решения в текстовой форме

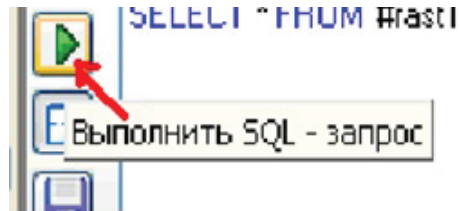


Рис. 7.52. Исполнение SQL-запроса



Рис. 7.53. Сохранение результатов

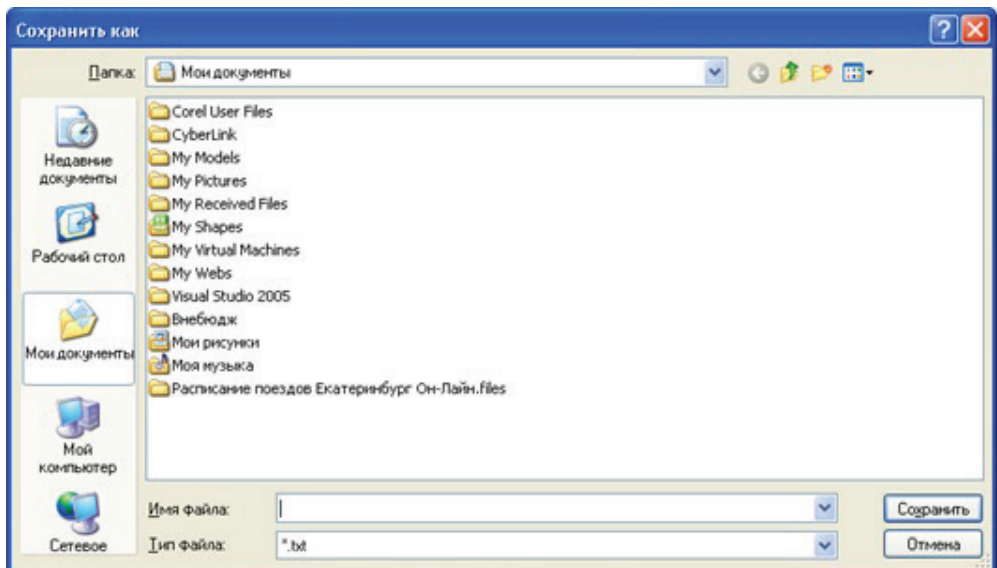


Рис. 7.54. Выбор пути сохранения результата

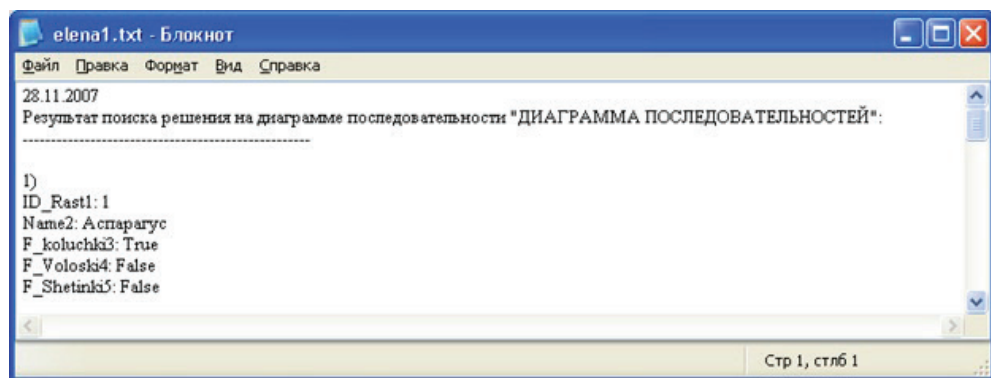


Рис. 7.55. Просмотр результатов поиска через текстовый документ

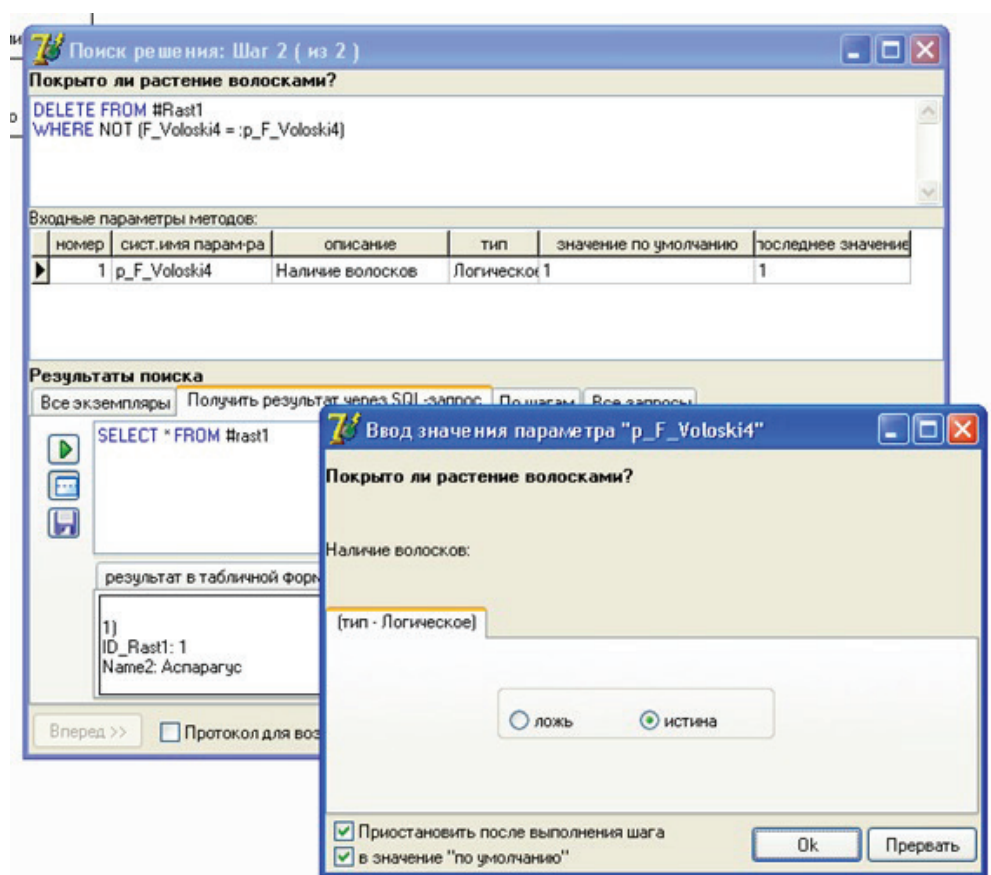


Рис. 7.56. Задание параметров запроса

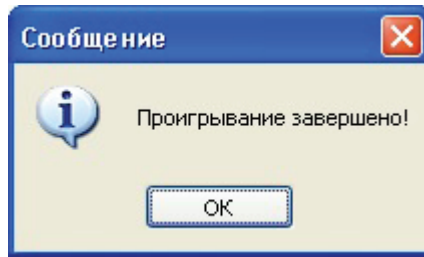


Рис. 7.57. Системное сообщение

Результаты проигрывания всей модели просматриваются аналогичным образом, что и на каждом шаге операции поиска решения.

8. Экспертное оценивание систем

В процессе проектирования информационных систем, на разных этапах необходимо принимать решения, выбирая из вариантов, которые имеют свои достоинства и недостатки. В некоторых случаях участники разработки делают выбор, основываясь на положительном опыте использования каких-то решений. Иногда требуется привлечение экспертов — специалистов в определенной области. Методы экспертных оценок — это методы организации работы со специалистами-экспертами и обработки мнений экспертов [15].

Экспертное оценивание как самостоятельная наука появилась в 1950—1960-х гг. Это связано с решением сложных научно-технических задач и попытками прогноза развития социальных систем. В тот период появились первые методы экспертных оценок. Далее кратко рассмотрим общие вопросы экспертного оценивания систем.

8.1. Общая методика проведения экспертного оценивания

В начале организации и проведения экспертизы принимается решение о формировании экспертной комиссии. Ее состав зависит от широты решаемой задачи: чем шире вопрос, тем больше различных специалистов необходимо привлекать. Ниже приводятся основные требования к экспертам:

- незаинтересованность в результате — очень важная характеристика качества эксперта, которая влияет на итог экспертизы;
- высокая квалификация в определенной области знаний — в экспертной комиссии могут быть люди с разной компетентностью, поэтому для количественной оценки степени важности мнения эксперта используют коэффициент компетентности;
- креативность мышления — способность предлагать нестандартные или принципиально новые решения — важное качество для ИТ-сферы;

- конструктивность мышления — способность давать практические решения с учетом реальных возможностей предприятия;
- психологическая независимость — эксперт должен отстаивать свою точку зрения, не поддаваясь влиянию авторитетов, свободно чувствовать себя в коллективных дискуссиях;
- самокритичность — умение учитывать мнение других экспертов, признавать свои ошибки.

При определении количества необходимых экспертов следует учитывать, что, с одной стороны, большое количество экспертов увеличивает достоверность полученных результатов, но с другой стороны, это затратное как по времени, так и по финансам мероприятие.

Все методы экспертных оценок делятся:

- на индивидуальные экспертные оценки — каждый эксперт дает свою личную оценку рассматриваемого вопроса в ходе анкетирования, интервью или тестирования; целесообразно использовать для решения узкого вопроса высококвалифицированного специалиста;
- коллективные экспертные оценки — получение обобщенного мнения группы экспертов с помощью определенного метода.

При проведении коллективной экспертной оценки используют следующие методы: метод комиссии, или круглого стола, мозговой штурм, метод Дельфи, метод построения дерева целей и деловую игру.

В основе всех экспертных методов лежит следующий порядок действий [15]:

- определение цели проведения экспертного оценивания;
- выбор метода проведения экспертного опроса (возможные методы будут описаны далее);
- подготовка раздаточного материала, содержащего имеющуюся информацию о проблеме, для экспертов оценивания. Если предполагается использование индивидуальных экспертных оценок, то на данном этапе готовят вопросы для анкетирования или интервьюирования;
- создание экспертной группы. Как описано выше, данный этап включает определение требований к экспертам, размеру группы и оценку их компетентности;
- проведение исследования с использованием выбранного метода;
- обработка результатов работы экспертов, которая включает в себя определение согласованности мнений экспертов, достоверности экспертных оценок, подготовку отчета, содержа-

щего цель исследования, описание экспертной группы и полученные выводы.

Для того чтобы результаты проведения экспертного оценивания были полезны для дальнейшего использования, полученная информация должна быть представлена в форме, удобной для последующего анализа. Эта задача достаточно простая, если альтернативные варианты могут быть оценены одним числом, например стоимость разработки и внедрения информационной системы в рублях. Если такой вариант не подходит, то выделяют важные показатели каждой альтернативы. Эксперты определяют степень важности каждого из них и проводят оценку отдельных признаков. Например, для выбора системы проведение видеоконференций, варианты можно оценивать по количеству одновременно подключенных пользователей, возможности использования бесплатного варианта системы и возможности записи и сохранения конференции.

Как видно из приведенных выше критериев, не все варианты могут быть оценены количественно (в рублях, штуках и т. п.). Иногда оценка носит качественный характер. В этом случае используются косвенные оценки. Эксперту предлагается сравнить характеристики альтернативы, приписав каждой из них определенное число, т. е. сформировать систему предпочтений, используя какую-то шкалу. Шкала — это инструмент (принятая система правил) оценки (измерения) каких-либо объектов или явлений. Рассмотрим четыре типа шкал [15].

- Номинальная шкала — это шкала, классифицирующая по названию. Это самый простой тип измерения. Номинальная шкала состоит из названий, типов, категорий или других признаков классификации по какому-то признаку. Примером измерения по номинальной шкале может служить оценка характеристики «возможности использования бесплатного варианта системы». В этом случае эксперт присваивает значение «да», равное единице, или «нет», равное нулю. Пример более сложной номинальной шкалы — языки программирования, которые могут быть использованы при разработке.

Очевидно, что данная шкала допускает только две операции: равно и неравно.

- Порядковая шкала — это шкала с цифрами, которые присваиваются вариантам для упорядочения. Результатом оценки является решение о том, что какая-либо характеристика альтернативы предпочти-

тельнее другой, никаких количественных оценок «на сколько лучше» эти цифры не дают. Например, шкала оценки ущерба при нарушении информационной безопасности: высокий, умеренный, низкий.

Данная шкала допускает следующие операции: равно, неравно, больше, меньше.

Номинальная и порядковая шкалы используются для оценки качественных признаков. Для оценки количественных характеристик используются интервальная шкала, шкала отношений, абсолютная шкала.

- Интервальная шкала — это шкала с цифрами, которые не только позволяют упорядочить оценки, но и определить, на сколько одна предпочтительнее другой. Нулевая точка и единица измерения выбираются при этом произвольно.

Пример интервальной шкалы — количество одновременно работающих пользователей в информационной системе: менее 50, 50–250, более 250.

Оценка удобства пользовательского интерфейса проводится по 10-балльной шкале, где 0 — абсолютно неудобно, 10 — очень удобно.

Данная шкала допускает следующие операции: равно, неравно, больше, меньше, сложение и вычитание.

- Шкала отношения — шкала с известной истинной нулевой точкой, известны отношения между значениями шкалы, нет единственной единицы измерения. Например, при помощи такой шкалы эксперты могут оценить стоимость проекта по разработке информационной системы в рублях.

Данная шкала допускает следующие операции: равно и неравно, больше и меньше, сложение и вычитание, умножение и деление.

- Абсолютная шкала — шкала с известной нулевой точкой и единой единицей измерения, шкала значений, у которой используются натуральные или действительные числа. Доступны все математические операции.

В зависимости от существа исследуемых объектов, для их оценки могут быть использованы различные шкалы.

Выбор шкалы зависит от характеристики измеряемой величины (может она быть оценена количественно или нет) и от целей исследования. Выше приведен пример оценки количества одновременно работающих пользователей в информационной системе по интервальной шкале. В данном случае для оценки мощности оборудования, на котором

будет работать система, интересует не абсолютное значение, а попадание в определенный интервал. Если информация о количестве пользователей необходима для организации для них рабочих мест, то следует использовать абсолютную шкалу — сообщить конкретное число.

Далее рассмотрим способы формирования экспертных оценок с помощью порядковой или интервальной шкалы, поскольку они наиболее часто применяются в экспертном оценивании. Характеристики, которые имеют количественные показатели, часто можно рассчитать без привлечения экспертов.

- Ранжирование — это расположение вариантов в порядке возрастания или убывания какой-либо характеристики. Ранжирование позволяет выбрать из исследуемой совокупности факторов наиболее существенный.

Результатом проведения ранжирования является ранжировка.

Если имеется n объектов, то в результате их ранжирования j -м экспертом, каждый объект получает оценку x_{ij} — ранг, приписываемый i -му объекту j -м экспертом.

Значения x_{ij} находятся в интервале $1-n$. Ранг самого важного фактора равен единице, наименее значимого — числу n .

Суммарный ранг объекта равен сумме рангов для каждого объекта. Результирующий ранг зависит от упорядочения суммарных рангов — меньший ранг получает объект с наименьшим суммарным рангом.

Достоинством метода является его простота, а недостатком — ограниченные возможности использования. При оценке большого количества объектов, экспертам очень трудно строить ранжированный ряд, поскольку приходится учитывать множество сложных связей. От этого недостатка свободен следующий метод.

В табл. 8.1 приведен пример оценки двумя экспертами квалификации трех команд разработчиков программного обеспечения.

Таблица 8.1

Оценка квалификации команд разработчиков методом ранжирования

Команда	1	2	3
1-й эксперт	2	1	3
2-й эксперт	3	1	2
Суммарный ранг	5	2	5
Результирующий ранг	2	1	2

Примечание. 1 — наивысшая квалификация, 3 — худшая квалификация.

- Парное сравнение — это установление предпочтения объектов при сравнении всех возможных пар. В этом методе необходимо в каждой из пар выявить более значимый объект или установить их равенство.

Метод парного сравнения подходит для большого количества объектов, а также в ситуациях незначительной разницы между объектами.

Для удобного использования этого метода составляется матрица размером $n \times n$, где n — количество сравниваемых объектов. На главной диагонали матрицы ставятся прочерки. При сравнении объектов матрица заполняется элементами a_{ij} следующим образом: если объект i предпочтительнее объекта j , то $a_{ij} = 1$, а $a_{ji} = 0$; если установлено равенство объектов, то обоим значениям присваивается 0.

Сумма по строке позволяет оценить относительную значимость объектов. Тот объект, для которого сумма окажется наибольшей, может быть признан наиболее важным (значимым).

В табл. 8.2 приведен пример оценки эксперта важности шести характеристик проекта по модернизации существующей информационной системы.

Таблица 8.2

Оценка характеристик проекта методом попарного сравнения

Характеристики	1	2	3	4	5	6	Σ
1	—	0	0	0	0	0	0
2	1	—	0	0	1	0	2
3	1	1	—	0	1	0	3
4	1	0	1	—	1	0	3
5	0	0	0	0	—	0	0
6	0	0	0	1	0	—	1

Эксперт считает третью и четвертую характеристики проекта наиболее значимыми.

- Непосредственная оценка. Часто бывает желательным не только упорядочить объекты, но и определить, на сколько один фактор более значим, чем другие. В этом случае диапазон изменения характеристик объекта разбивается на отдельные интервалы, каждому из них приписывается определенная оценка (балл), например 0—10. Поэтому метод непосредственной оценки иногда называют балльным методом.

Смысл метода состоит в том, что эксперт помещает каждый из анализируемых объектов в определенный интервал (приписывает балл).

Измерителем при этом является степень обладания объекта тем или иным свойством.

Число интервалов, на которые разбивается диапазон изменения свойства, может быть различным для разных экспертов. Кроме того, метод разрешает давать одну и ту же оценку (т. е. помещать в один и тот же интервал) различным объектам. Например, применение метода непосредственной оценки для предыдущего примера «Оценка характеристик проекта» представлено в табл. 8.3.

Таблица 8.3

Сравнение характеристик проекта методом непосредственной оценки

Оценки	Характеристики					
	1	2	3	4	5	6
Баллы	30	65	80	85	20	50
Ранги	4	2	1	1	5	3

Далее рассмотрим действия, выполняемые на этапе обработки результатов опроса. При обработке индивидуальных экспертных оценок необходимо из индивидуальных оценок получить обобщенную. Для этого используется метод согласования оценок, который имеет различные варианты [15]. Например, нахождение медианы — оценки, по отношению к которой число больших оценок равняется числу меньших. Обобщенная оценка для группы экспертов может вычисляться как среднее арифметическое.

В некоторых случаях при анализе результатов экспертизы следует учесть неравнозначность объектов или их характеристик. Для этого вводится вес каждого из них. Кроме того, следует оценить степень согласованности мнений экспертов. Конечно, все эксперты могут давать различные оценки по анализируемому вопросу. Если мнения экспертов сильно расходятся, то групповая оценка не имеет смысла. Разработаны методы оценки согласованности мнений экспертов [15].

Далее рассмотрим некоторые методы коллективной оценки.

8.2. Метод мозгового штурма

Мозговой штурм (мозговая атака, брейнсторминг) — коллективный метод принятия решений, который позволяет стимулировать творческую деятельность группы участников. Автором этого метода счита-

ется американец Алекс Осборн. Он заметил, что люди не хотят высказывать смелые и неожиданные идеи, поскольку боятся критики и высмеивания. Он описал метод, в котором этап генерирования идей отделен от процесса их обсуждения и оценки. Цель проведения мозгового штурма заключается в получении за ограниченное время большого количества решений обсуждаемой проблемы [15]. Существует несколько вариантов этого метода. Ниже описываются этапы проведения мозгового штурма одного из них.

- Формируется группа для обсуждения и выбирается ведущий дискуссии. Оптимальное для дискуссии число участников — 6–8 чел. Группа может быть разделена на генераторов идей и аналитиков, которые занимаются только обсуждением, или все участники могут предлагать решения.
- Четкое описание задачи для обсуждения, которое включает в себя цель, которую хотелось бы достигнуть в результате, факторы, которые этому препятствуют, возможные последствия бездействия и др.
- Разминка. В некоторых случаях требуется снять закрепощенность участников мозгового штурма и настроить их на творческую волну. Для этого проводят разминку: играют в ассоциации или обсуждают вопрос, близкий к главной теме.
- Этап генерирования идей, или «мозговая атака». Ведущий старается, чтобы высказались все участники. На данном этапе нужно получить как можно больше решений по заявленной проблеме, каждая из них должна быть зафиксирована. При этом соблюдаются следующие правила:
 - ✓ предлагаемые идеи не имеют авторов;
 - ✓ предлагаемые идеи нельзя критиковать;
 - ✓ на данном этапе предложенные решения не анализируются, поэтому высказываются даже идеи, для которых сложно сохладу придумать практическую реализацию;
 - ✓ поощряется развивать и комбинировать уже предложенные идеи;
 - ✓ за одно выступление можно предложить только одно решение;
 - ✓ количество выступлений одного участника никак не ограничено.
- Анализ предложений. На данном этапе работают критики, если изначально участники разделены на две группы. Данный этап

можно проводить сразу после предыдущего этапа, а можно дать некоторое время (до недели) на осмысление предложенных вариантов. На этапе анализа предложений из всего списка выбирают наиболее удачные. При этом можно использовать правило Парето «20:80», т. е. отбираем 20 % идей. Затем можно провести сокращенный мозговой штурм для развития этих идей или провести анализ их реализации.

Можно выделить следующие достоинства этого метода:

- ✓ простой способ проведения;
- ✓ позволяет совершенствовать представленные идеи и на основе их разрабатывать новые.

К недостаткам можно отнести следующие моменты:

- ✓ мероприятие будет неэффективным, если не все участники будут высказывать идеи;
- ✓ несмотря на правила проведения «мозговой атаки», некоторые участники могут невербально демонстрировать свое критическое отношения к высказываниям.

Этих недостатков лишен следующий метод — метод Дельфи.

8.3. Метод Дельфи

Метод Дельфи — метод экспертных оценок, отличающийся отсутствием коллективных обсуждений. Данный метод изначально разрабатывался в целях научно-технического прогнозирования будущего. В его основе лежит идея получения групповой оценки не только путем усреднения результатов отдельных экспертов, но и корректировки их мнений на основе остальных в процессе экспертизы [15].

Перечислим основные особенности метода Дельфи:

- анонимность экспертов;
- получение независимых результатов от каждого участника, отсутствие коллективных обсуждений;
- процедура опроса в несколько этапов;
- получение экспертами обратной связи.

Применение метода Дельфи удобно в тех случаях, когда сложно собрать экспертов в одном месте из-за большой территориальной удаленности или сложного графика для проведения, например, «мозгового

штурма». Кратко опишем порядок проведения экспертного оценивания с использованием этого метода.

- Формирование группы экспертов и группы аналитиков. Ни один эксперт не знает, кто еще включен в группу. Иногда исследуемый вопрос достаточно узок, а список возможных экспертов всем очевиден. В соответствии с правилами метода экспертам нельзя обмениваться мнениями по исследуемому вопросу. Аналитики занимаются обработкой результатов работы экспертов.
- Участникам экспертной группы сообщается вопрос или задача для обсуждения. Каждый из них самостоятельно разбивает его (ее) на более мелкие. Результаты их работы поступают аналитикам, которые выбирают наиболее повторяющиеся и актуальные. На основании выбранной информации составляют анкету с вопросами, который рассылается экспертам.
- Эксперты заполняют анкету, подкрепляя свои ответы аргументами: наличие необходимых для решения ресурсов, оценка его эффективности. Также эксперты могут запросить дополнительную информацию по проблеме. Ответы экспертов опять обрабатываются аналитиками. Составляется новая анкета, которая также включает анонимные ответы экспертов с предыдущего этапа. Новая анкета рассылается экспертам. Эксперты опять отвечают на вопросы, при этом они могут изменить свою точку зрения в результате знакомства с ответами других. Этот процесс продолжается до тех пор, пока аналитики не получают согласованного мнения экспертов.
- На заключительном этапе на основе результатов работы экспертов вырабатывается окончательное решение, даются практические рекомендации по его реализации.

Из описанного выше алгоритма проведения экспертного оценивания методом Дельфи вытекают его достоинства:

- ✓ исключение влияния мнения авторитетов на других участников за счет анонимности и заочного проведения;
- ✓ возможность изменить свое мнение без негативных последствий для эксперта;
- ✓ устранение конфликтных ситуаций — эксперты не встречаются друг с другом.

К недостаткам этого метода можно отнести:

- ✓ процесс экспертизы достаточно длительный по времени;

- ✓ отсутствие открытой дискуссии может привести к злоупотреблениям со стороны группы аналитиков, приводящим к искажению конечного результата.

8.4. Метод сценариев

Метод сценариев — методов экспертных оценок, с помощью которого можно получить варианты развития ситуации при принятии определенных решений, влияние различных факторов на ситуацию. Под сценарием понимают документ, содержащий анализ рассматриваемой проблемы и разные варианты ее решения. Сценарий позволяет выявить риски каждого из вариантов.

Применение метода сценариев актуально на начальном этапе крупного по инвестициям и длительного по реализации ИТ-проекта. На начальном этапе необходимо провести анализ внешней и внутренней среды и выявить факторы, которые будут оказывать влияние на разработку информационной системы.

На этапе написания сценария проводится анализ влияния этих факторов на проект, предлагаются различные варианты решения возникающих проблем, можно рассматривать совместное влияние нескольких факторов. В результате анализа отбираются факторы, которые оказывают наибольшее как положительное, так и отрицательное влияние на проект, описываются последствия их влияния. Комбинирование факторов позволяет оценить взаимосвязи и зависимости между возможными результатами действия фактов. В итоге пишутся сценарии развития ИТ-проекта, например один оптимистический, другой пессимистический, а третий нейтральный [16]. Каждый сценарий должен быть логичен и непротиворечив [16].

Работа экспертов может быть построена одним из нескольких способов, например, получение согласованного мнения с использованием метода Дельфи или повторяющаяся процедура независимых сценариев — написание независимых сценариев для каждого выделенного фактора, а затем последовательное согласование объединение сценариев.

Результаты метода сценариев используются при реализации ИТ-проекта: выбирают наиболее благоприятное решение, стараются учесть или минимизировать возможные негативные влияния факторов.

8.5. Метод морфологического анализа

Метод морфологического анализа — метод получения оригинальных решений, который основан на выявлении морфологических признаков — важных составляющих решаемой задачи — и составлении всех возможных сочетаний их реализаций [15]. Этот метод разработан астрофизиком Фрицем Цвикки в 1930-е гг., в своем методе автор старался систематически охватить все варианты решения, исключив влияние случайности.

Морфологический метод состоит из следующих этапов:

- формулировки проблемы исследования;
- определения морфологических признаков — структурных элементов, из которых состоит объект, или элементарных действий, если обсуждаются способы решения задачи;
- определения возможных вариантов значений для каждого морфологического признака;
- построения морфологического ящика — многомерной матрицы; каждый индекс определяет морфологический признак, элемент матрицы — возможный вариант их комбинации;
- анализа всех возможных комбинаций морфологических признаков. Такой подход заставляет рассмотреть все сочетания выбранных признаков, иногда очень неожиданные, которые могут привести к нестандартным решениям;
- выбора оптимального варианта.

Выделим следующие достоинства морфологического анализа:

- равноценность всех выделенных признаков объекта или способа решения задачи;
- рассмотрение всех комбинаций признаков без ограничений;
- возможность выявления нестандартного подхода к решаемой проблеме.

К недостаткам можно отнести:

- сложность выявления существенных и независимых признаков проблемы;
- большую трудоемкость для анализа всех возможных решений при большом количестве признаков.

Метод морфологического анализа можно применять, например, при проектировании пользовательского интерфейса. В качестве мор-

фологических признаков можно выделить: способ навигации, способ сообщений о состоянии системы, организацию помощи пользователю. Перечислим все морфологические признаки и их возможные значения (табл. 8.4).

Таблица 8.4

Морфологические признаки и их значения

Параметр	Значение 1	Значение 2	Значение 3
Способ навигации	Главное меню	Дерево команд	Лента
Способ сообщений о состоянии системы	Строка состояния внизу формы	Всплывающие окна с сообщениями	Сообщения в заголовке формы
Помощь	Отсутствует	Всплывающие подсказки	Кнопка Помощь

Необходимо проанализировать все получившиеся комбинации и выбрать оптимальный вариант. В качестве критериев отбора можно использовать эвристические правила проектирования пользовательского интерфейса.

8.6. Экспертные игры

Игра — вид человеческой деятельности, который позволяет промоделировать другие ее виды. Если участниками игры являются эксперты, то такая игра называется экспертной. Использование экспертных игр в качестве метода экспертной оценки заключается в проведении эксперимента. При этом участникам предлагается определенная ситуация, а эксперты должны предложить решение, используя общие и специальные знания. В начале игры определяются ее правила и схема проведения. В конце проводится анализ игры и оценка полученного решения на соответствие цели исследования. Определенную трудность при подготовке игры создает степень детализации ситуации. Слишком большое количество деталей приведет к усложнению игры, эксперты вынуждены будут учитывать всю информацию и получают неэффективное решение. Если описание ситуации слишком упрощенное, то полученное решение будет достаточно общим и мало полезным. Рекомендуется при составлении описания ситуации постоянно учиты-

вать цель исследования и не включать детали, которые не будут использованы [16].

При разработке информационных систем, экспертные игры можно использовать на начальном этапе проекта. Например, для лучшего понимания бизнес-процессов можно предложить сотрудникам компании проиграть какие-то стандартные ситуации. Для оценки конкурентов можно предложить специалистам промоделировать действия конкурентов при появлении на предприятии новой информационной системы.

8.7. Применение методов экспертных оценок при разработке ИТ-проектов

Разработка информационных систем связана с разнообразными рисками. При эксплуатации системы также возникают риски, связанные с обеспечением непрерывности автоматизированных бизнес-процессов: постоянная работоспособность программного обеспечения, сохранность данных и т. п. Для управления рисками можно применять методы экспертных оценок, также они применимы при принятии решений в условиях неопределенности данных.

Поскольку разработка или покупка информационной систем — финансово затратный проект как на начальном этапе, так и в процессе эксплуатации, то на этапе планирования необходимо оценить эффективность ИТ-решения с точки зрения реализации действительных потребностей бизнеса, уровня конкуренции на рынке. Если расходы на разработку или покупку новой системы оценить реально, то эффект от ее внедрения не всегда поддается четкой оценке из-за отсутствия статистических данных и надежных методов. Определенную сложность вносят непрогнозируемые изменения в бизнес-процессах предприятия, связанные с политической и экономической обстановкой в стране. Очевидные преимущества от внедрения информационной системы, которые обычно называют снижением трудовых и временных затрат, минимизацией ошибок, связанных с человеческим фактором, увеличением конкурентоспособности, можно оценить качественно, но сложно перевести в количественные оценки. Все это указывает на целесообразность применения методов экспертной оценки для обо-

снования преимущества разработки или покупки новой информационной системы или выбора альтернативного распределения ресурсов для решения задач бизнеса с оценкой их преимуществ и недостатков. В зависимости от вопросов, которые предполагается оценить, в качестве экспертов выбирают:

- руководство и менеджмент компании — эти лица принимают решение о разработке или покупке информационной системы, формулируют цели ИТ-проекта, упорядочивают их по степени важности;
- пользователей системы, как внутренних — сотрудников предприятия, так и внешних — клиентов компании. Эти лица владеют информацией о реальных процессах компании, могут оценить полезность и удобство предлагаемых решений.

На этапе анализа требований к информационной системе и моделировании бизнес-процессов, в качестве экспертов следует привлекать специалистов в предметной области, которые хорошо разбираются в технологических процессах предприятия.

На этапе проектирования информационной системы или выбора конфигурации готовой необходимо провести анализ соответствия отдельных частей системы требованиям бизнес-процессов предприятия. Привлечение экспертов в проектирование систем позволит обеспечить поддержку принятия решения при выборе вариантов с оценкой их предпочтительности.

На этапах разработки и внедрения экспертное оценивание поможет при решении задач в условиях ограничения по времени и в критических ситуациях.

Библиографический список

1. Dennis, A. Systems analysis and design / Alan Dennis, Barbara Haley Wixom, Roberta M. Roth. — 5th ed. — Wiley : [Б. и.], 2012. — 363 p. — ISBN 978-1-1180576-2-9.
2. Дегтярев, М. Г. Логика : учеб. для студ. высш. учеб. заведений / М. Г. Дегтярев, Е. К. Войшвилло. — Москва : ВАЛДОС-ПРЕСС, 2001. — 528 с. — ISBN 5-305-00001-7.
3. Вендров, А. М. CASE-технологии. Современные методы и средства проектирования информационных систем / А. М. Вендров. — Москва : Финансы и статистика, 1998. — 175 с. — ISBN 5-279-01979-8.
4. Системный анализ и принятие решений : словарь-справочник / под ред. В. Н. Волковой, В. Н. Козловой. — Москва : Высшая школа, 2004. — ISBN 5-06-004875-6.
5. Методы определения экономического эффекта от ИТ-проекта. — URL: https://iteam.ru/publications/it/section_53/article_2905 (дата обращения: 25.01.2021). — Загл. с титул. экрана.
6. Strategic Alignment Электронный ресурс. — URL: <http://www.hr-portal.ru/varticle/strategic-alignment-strategicheskoe-vyravnivanie-venkatraman> (дата обращения: 25.01.2021). — Загл. с титул. экрана.
7. Симонов, А. П. Архитектурный подход к выравниванию бизнеса и ИТ в организации / А. П. Симонов. — URL: <http://samag.ru/archive/article/3303> (дата обращения: 25.01.2021). — Загл. с титул. экрана.
8. Henderson J. Strategic Alignment: Leveraging information technology for transforming organizations / J. Henderson, N. Venkatraman // IBM Systems Journal. — 1993. — Vol. 32, no 1. — P. 4–16.
9. Мацяшек, Лешек. Анализ требований и проектирование систем. Разработка информационных : пер. с англ. / Лешек Мацяшек. — Москва : Вильямс, 2002. — 432 с. — ISBN 5-84590-2762.

10. Что такое бенчмаркинг: примеры удачного использования <http://hardcorecase.ru/data/termin/benchmarking.html> (дата обращения: 25.01.2021). — Загл. с титул. экрана.
11. Федоров, И. Г. Моделирование бизнес-процессов в нотации BPMN 2.0 / И. Г. Федоров ; М-во образования и науки Российской Федерации, Московский гос. ун-т экономики, статистики и информатики (МЭСИ). — Москва : МЭСИ, 2013. — 264 с. — ISBN 978-5-7764-0772-7.
12. Спицина, И. А. Применение системного анализа при разработке пользовательского интерфейса информационных систем: учебное пособие для студентов вуза, обучающихся по направлению подготовки 09.03.01 «Информатика и вычислительная техника» / И. А. Спицина, К. А. Аксенов ; науч. ред. Л. Г. Доросинский ; Министерство образования и науки Российской Федерации, Уральский федеральный университет имени первого Президента России Б. Н. Ельцина. — Екатеринбург : Изд-во Урал. ун-та, 2018. — 100 с. — ISBN 978-5-7996-2265-7.
13. Разработка CASE-средства на основе интеграции функционального и объектно-ориентированного моделирования / К. А. Аксенов, Е. К. Боярчук, Е. А. Чарина, И. А. Спицина // X Отчетная конференция молодых ученых ГОУ ВПО УГТУ-УПИ. — Екатеринбург : ГОУ ВПО УГТУ-УПИ, 2006. — С. 285–288.
14. Aksyonov, K. Identification of Problems and Limitations of the Planning of Fuel Distribution through the Gas Station Network and their Solution / K. Aksyonov, O. Aksyonova, H. Ayvazyan // SIBIRCON 2019 — International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings. — P. 647–651. — DOI: 10.1109/SIBIRCON48586.2019.8958191.
15. Орлов, А. И. Менеджмент : учебник / А. И. Орлок. — Москва : Изумруд, 2003. — 298 с.
16. Могилина, В. А. Применение метода сценариев для разработки стратегических альтернатив развития угледобывающих предприятий // НАУКОВЕДЕНИЕ. — Т. 7, № 1 (2015). — URL: <http://naukovedenie.ru/PDF/89EVN115.pdf> (доступ свободный). — Загл. с экрана. — Яз. рус., англ. — DOI: 10.15862/89EVN115.
17. Молин, А. С. Исследование системы управления : учеб. для вузов / А. С. Молин, В. И. Мухин. — Москва : МГУ, 2002. — 329 с. — ISBN 5-7598-0192-9.

Учебное издание

Спицина Ирина Александровна
Аксенов Константин Александрович

**СИСТЕМНЫЙ АНАЛИЗ
И МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

Редактор И. В. Меркурьева
Верстка О. П. Игнатевой

Подписано в печать 15.03.2021. Формат 70×100/16.
Бумага офсетная. Цифровая печать. Усл. печ. л. 9,5.
Уч.-изд. л. 6,1. Тираж 30 экз. Заказ 17.

Издательство Уральского университета
Редакционно-издательский отдел ИПЦ УрФУ
620049, Екатеринбург, ул. С. Ковалевской, 5
Тел.: +7 (343) 375-48-25, 375-46-85, 374-19-41
E-mail: rio@urfu.ru

Отпечатано в Издательско-полиграфическом центре УрФУ
620083, Екатеринбург, ул. Тургенева, 4
Тел.: +7 (343) 358-93-06, 350-58-20, 350-90-13
Факс: +7 (343) 358-93-06
<http://print.urfu.ru>

